

JOSÉ LUIZ SILVEIRA

**JOGO EDUCACIONAL PARA APOIAR O ENSINO DE MELHORIA
DE PROCESSO DE SOFTWARE COM FOCO NO NÍVEL G DE
MATURIDADE DO MPS.BR**

São José (SC), Agosto de 2012



UNIVERSIDADE DO VALE DO ITAJAÍ
CURSO DE MESTRADO ACADÊMICO EM
COMPUTAÇÃO APLICADA

**JOGO EDUCACIONAL PARA APOIAR O ENSINO DE
MELHORIA DE PROCESSO DE SOFTWARE COM FOCO NO
NÍVEL G DE MATURIDADE DO MPS.BR**

por

José Luiz Silveira

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Computação
Aplicada.

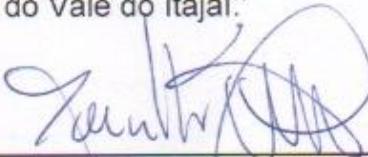
Orientador: Marcello Thiry, Dr.

São José (SC), Agosto de 2012

**"JOGO EDUCACIONAL PARA APOIAR O ENSINO DE MELHORIA
DE PROCESSO DE SOFTWARE COM FOCO NO NÍVEL G DE
MATURIDADE DO MPS.BR".**

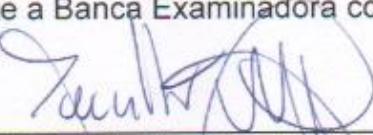
José Luiz Silveira

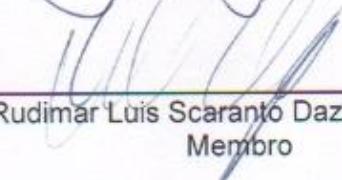
'Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Computação Aplicada, Área de Concentração Computação Aplicada e aprovada em sua forma final pelo Programa de Mestrado Acadêmico em Computação Aplicada da Universidade do Vale do Itajaí.'


Prof. Marcello Thiry Comicholi da Costa, Dr.
Orientador


Prof. Cesar Albenes Zeferino, Dr.
Coordenador do Programa Mestrado em Computação Aplicada

Apresentado perante a Banca Examinadora composta pelos Professores:


Prof. Marcello Thiry Comicholi da Costa, Dr.
(UNIVALI)
Presidente


Prof. Rudimar Luis Scaranto Dazzi, Dr. (UNIVALI)
Membro


Profa. Fabiane Barreto Vavassori Benitti, Dra. (UNIVALI)
Membro


Prof. Clênio Figueiredo Salviano, Dr. (CTI)

Membro

Itajaí (SC), 23 de agosto de 2012.

*Dedico este trabalho a meus pais, Luiz Silveira e Vera Lúcia Conrat Silveira, e a minha namorada
Aline Mannes Andrade.*

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois sem ele nada disso seria possível.

Agradeço a meu orientador, Dr. Marcello Thiry, pelas imensas contribuições para o desenvolvimento deste trabalho.

Agradeço também aos membros da banca deste trabalho.

Agradeço a meus pais, Luiz Silveira e Vera Lúcia Conrat Silveira, por sempre terem me incentivado a estudar.

Agradeço a minha namorada, Aline Mannes Andrade, pela paciência, apoio, incentivo, e por estar do meu lado nos momentos mais difíceis.

Agradeço também a Idalete Mannes Andrade e José Carlos Andrade, pelo apoio e incentivo.

Agradeço ao Vinícius Roberto, André Roberto, e Rúbia Kikuchi, pelo desenvolvimento e design do jogo.

Agradeço ao Narcizo Thiago pela ajuda na construção da base de conhecimento do consultor.

Agradeço ao Eduardo Luz e Ricardo Barbi, pelo apoio e por permitirem que eu me ausentasse do trabalho quando precisei.

Agradeço a todos os meus colegas de trabalho pelo apoio quando eu precisei me ausentar.

Agradeço ao Gilberto Pacheco pelo apoio na elaboração dos diálogos do jogo.

Agradeço a todos os participantes dos experimentos realizados.

Agradeço aos meus colegas de mestrado, Daniela Souza Moreira, Djoni Antônio da Silva e Leonardo Tristão, pelo apoio e parceria durante a realização do mestrado.

Agradeço a todos meus amigos e familiares por entenderem quando eu não pude estar presente.

E por fim, agradeço também ao Fundo de Apoio à Manutenção e ao Desenvolvimento da Educação Superior (FUMDES), programa de concessão de bolsas de estudo da Secretaria de Educação do Estado de Santa Catarina, por ter concedido a bolsa para a realização deste trabalho.

JOGO EDUCACIONAL PARA APOIAR O ENSINO DE MELHORIA DE PROCESSO DE SOFTWARE COM FOCO NO NÍVEL G DE MATURIDADE DO MPS.BR

José Luiz Silveira

Abril / 2012

Orientador: Marcello Thiry, Dr.

Área de Concentração: Computação Aplicada

Linha de Pesquisa: Engenharia de Software

Palavras-chave: Melhoria de Processo de Software, Engenharia de Software, Jogos Educacionais.

Número de páginas: 243

RESUMO

Embora o mercado mundial de software esteja entre os de maior crescimento, atualmente, pesquisas evidenciam que as empresas do setor ainda continuam enfrentando problemas com prazo, custo, qualidade e satisfação de seus clientes. Neste sentido, modelos de capacidade de software têm sido desenvolvidos para apoiar as organizações na melhoria de seus processos e a respectiva redução dos problemas identificados. Entretanto, existem desafios para a adoção destes modelos. Os modelos são genéricos e não apresentam como eles devem ser implementados, deixando que cada organização faça sua própria interpretação. Desta forma, a capacitação adequada nestes modelos é um fator chave de sucesso para iniciativas de melhoria. Muitas vezes, treinamentos teóricos são insuficientes para o real entendimento do modelo e atividades que simulem situações do mundo real podem aumentar a fixação nos conceitos e aplicação do modelo. Assim, neste trabalho foi desenvolvido um jogo educacional que simula o cenário de uma empresa de desenvolvimento de software com situações em que os resultados esperados do nível G de maturidade do MPS.BR são explorados. O jogo foi avaliado por meio de experimentos que permitiram afirmar que o jogo auxilia na aprendizagem de melhoria de processo de software.

EDUCATIONAL GAME TO SUPPORT THE TEACHING OF SOFTWARE PROCESS IMPROVEMENT WITH FOCUS ON THE G LEVEL OF MATURITY OF MPS.BR

José Luiz Silveira

08 / 2012

Advisor: Marcello Thiry, Dr.

Area of Concentration: Applied Computer Science

Research Line: Software Engineering

Keywords: Software Process Improvement, Software Engineering, Educational Game.

Number of pages: 243

ABSTRACT

Although the worldwide market for software is among the fastest growing, currently, research showed that companies in the industry are still experiencing problems with schedule, cost, quality and customer satisfaction. In this sense, software capability models have been developed to assist organizations in improving their processes and the corresponding reduction of the problems identified. However, there are challenges to adoption of these models. The models are generic and do not show how they should be implemented, allowing each organization to make their own interpretation. Thus, the proper training of these models is a key success factor for improvement initiatives. Often, theoretical training is insufficient for the real understanding of the model and activities that simulate real-world situations can increase the setting on the concepts and application of the model. In this work we developed an educational game that simulates the scenario of a software development company with situations in which the expected results of the G level of maturity of MPS.BR are explored. The game was evaluated by experiments which show that the game helps in the learning of software process improvement.

LISTA DE ILUSTRAÇÕES

Figura 1. Estrutura básica do CMMI-DEV versão 1.3	35
Figura 2. Modelo conceitual da ISO/IEC 15504.....	37
Figura 3. Categorias, Grupos de Processo e Processos da ISO/IEC 15504-5.....	41
Figura 4. Componentes do MPS.BR.....	44
Figura 5. <i>Framework</i> para desenvolvimento de jogos educacionais	56
Figura 6. Estrutura do modelo de avaliação de jogos educacionais.....	63
Figura 7. Quantidade trabalhos selecionados por fonte de pesquisa.....	74
Figura 8. Fluxo do jogo	83
Figura 9. Modelo de classes	103
Figura 10. Modelo Entidade-Relacionamento	104
Figura 11. Animação inicial do jogo.....	112
Figura 12. Tela de login e tela de cadastro de jogador	113
Figura 13. Menu principal do jogo.....	113
Figura 14. Tela de regras do jogo	114
Figura 15. Algumas das telas do tutorial do jogo	114
Figura 16. Tela de ranking de empresas do jogo	114
Figura 17. Tela de listagem das empresas do jogador	115
Figura 18. Tela principal do jogo.....	116
Figura 19. Contratar desenvolvedor	117
Figura 20. Treinar desenvolvedor	117
Figura 21. Recompensar desenvolvedor	118
Figura 22. Demitir desenvolvedor	119
Figura 23. Consultar situação do projeto	120
Figura 24. Reunião de acompanhamento de projeto	120
Figura 25. Visualizar tarefas	121
Figura 26. Tela de diálogo	122
Figura 27. Tela de feedback	122
Figura 28. Tela de aviso de resultados esperados atendidos	123
Figura 29. Tela de relatório de implementação.....	123
Figura 30. Tela de relatório de gastos e receitas	124

Figura 31. Tela final do jogo.....	124
Figura 32. Design de experimento para avaliação do jogo SPI City	129
Figura 33. Aspecto motivação	144
Figura 34. Aspecto experiência do usuário.....	146
Figura 35. Aspecto aprendizagem.....	147

LISTA DE TABELAS

Tabela 1. Áreas de processo, categorias e níveis de maturidade do CMMI-DEV	36
Tabela 2. Atributos de processo e níveis de capacidade da ISO/IEC 15504	39
Tabela 3. Níveis de maturidade do MR-MPS	45
Tabela 4. Conteúdos x atividade de aprendizagem x estilos de jogos	55
Tabela 5. Questões por dimensão do modelo de avaliação de jogos educacionais.....	66
Tabela 6. Pergunta de pesquisa do mapeamento sistemático.....	67
Tabela 7. Palavras chave utilizadas no mapeamento sistemático	68
Tabela 8. Termos de busca do mapeamento sistemático	69
Tabela 9. Quantidade de trabalhos retornados por fonte de pesquisa	73
Tabela 10. Jogos x resultados esperados.....	75
Tabela 11. Situações que farão a motivação diminuir ou aumentar	90
Tabela 12. Salário dos funcionários de acordo com seu papel	92
Tabela 13. Dicionário de dados da entidade empresa	104
Tabela 14. Dicionário de dados da entidade jogador	104
Tabela 15. Diálogos relacionados com cada resultado esperado, grau de cobertura e justificativa	108
Tabela 16. Sequência de quadros da animação inicial do jogo.....	111
Tabela 18. Notas do grupo experimental e controle no pré-teste e pós-teste.....	135
Tabela 19. Classificação pelas diferenças das notas do pré e pós-teste	136
Tabela 20. Cálculo de T1 e T2.....	136
Tabela 21. Tabela de valores críticos do teste U.....	137
Tabela 22. Classificação pelas notas do pós-teste.....	138
Tabela 23. Cálculo de T1 e T2.....	138
Tabela 24. Notas no pré-teste e pós-teste no experimento online.....	140
Tabela 25. Resultado quantitativo do experimento online	140
Tabela 26. Resultado quantitativo geral.....	142
Tabela 27. Análise quantitativa por nível de aprendizagem	142
Tabela 28. Síntese da reação dos participantes ao SPI City	149
Tabela 29. Síntese dos resultados da avaliação.....	150
Tabela 30. Informações extraídas dos trabalhos analisados no mapeamento sistemático	173
Tabela 31. Evento 1 - definição da forma de trabalho	199

Tabela 32. Evento 2 - definição do ciclo de vida.....	202
Tabela 33. Evento 3 - definição do que colocar no plano de projeto.....	208
Tabela 34. Evento 4 - pedido de alteração de requisitos.....	215
Tabela 35. Evento 5 - definição do planejamento de riscos.....	219
Tabela 36. Evento 6 - problema com os artefatos do projeto	223
Tabela 37. Questões do pós-teste por nível da taxonomia de Bloom	232
Tabela 38. Questões do pós-teste por nível da taxonomia de Bloom	236

LISTA DE ABREVIATURAS E SIGLAS

ADDIE	<i>Analysis Design Development Implementation and Evaluation</i>
AP	Atributo de Processo
CA	Consultores de Aquisição
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
CMMI-DEV	<i>Capability Maturity Model Integration for Development</i>
GDD	<i>Game Design Document</i>
GG	<i>Generic Goals</i>
GP	<i>Generic Practices</i>
GR	<i>Generic Resources</i>
GWP	<i>Generic Work Product</i>
IA	Instituições Avaliadoras
IDEAL	<i>Initiating Diagnosing Establishing Acting Learning</i>
IEC	<i>Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
II	Instituições Implementadoras
IOGE	Instituições Organizadoras de Grupo de Empresas
ISO	<i>International Organization for Standardization</i>
MA.MPS	Método de Avaliação de Melhoria do Processo de Software
MN.MPS	Método de Negócio de Melhoria do Processo de Software
MOF	<i>Meta-Model Facility</i>
MPE	Micro e Pequenas Empresas
MPS.BR	Melhoria do Processo de Software Brasileiro
MPS	Melhoria de Processo de Software
MR.MPS	Modelo de Referência de Melhoria do Processo de Software
OMG	<i>Object Management Group</i>
PA	<i>Process Attribute</i>
PB	Prática Base
PCP	Perfil de Capacidade de Processo
PDCA	<i>Plan Do Check Action</i>
PG	Prática Genérica
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
PRO2PI	<i>Process Capability Profile to drive Process Improvement</i>
PRO2PI-WORK	<i>Process Capability Profile to drive Process Improvement Establishing Workshop Method</i>
PT	Produto de Trabalho
PTG	Produto de Trabalho Genérico
RAP	Resultado de Atributo de Processo
RG	Recurso Genérico
RH	Recursos Humanos
RPG	<i>Role Playing Game</i>
SEI	<i>Software Engineering Institute</i>
SG	<i>Specific Goal</i>
SOFTEX	Sociedade para Promoção da Excelência do Software Brasileiro

SP
SPICE
TDD
UML
WP

Specific Practice
Software Process Improvement and Capability dEtermination
Technical Design Document
Unified Modeling Language
Work Product

SUMÁRIO

1 INTRODUÇÃO	16
1.1 PROBLEMA DE PESQUISA	17
1.1.1 Solução Proposta	20
1.1.2 Delimitação de Escopo	21
1.1.3 Justificativa	22
1.2 OBJETIVOS	23
1.2.1 Objetivo Geral.....	23
1.2.2 Objetivos Específicos	23
1.3 METODOLOGIA.....	24
1.3.1 Metodologia da Pesquisa.....	24
1.3.2 Procedimentos Metodológicos	25
1.4 ESTRUTURA DA DISSERTAÇÃO.....	26
2 FUNDAMENTAÇÃO TEÓRICA	28
2.1 MELHORIA DE PROCESSO DE SOFTWARE	28
2.1.1 Definições.....	28
2.1.2 Modelos de capacidade de processo	31
2.2 JOGOS	51
2.2.1 Definição	51
2.2.2 Características.....	52
2.2.3 Classificações	52
2.2.4 Jogos Educacionais.....	53
2.2.5 Projeto de Jogos Educacionais	55
2.2.6 Avaliação de Jogos Educacionais	57
3 MAPEAMENTO SISTEMATICO	67
3.1 PLANEJAMENTO.....	67
3.1.1 Perguntas de Pesquisa.....	67
3.1.2 Estratégia de Busca	67
3.1.3 Seleção dos Trabalhos	70
3.1.4 Estratégia de Extração dos Dados.....	72
3.2 RESULTADOS	73
3.2.1 Execução	73
3.2.2 Análise dos Resultados.....	75
3.3 DIFERENCIAL DO TRABALHO	76
4 DESENVOLVIMENTO	78
4.1 OBJETIVOS EDUCACIONAIS.....	78
4.2 GAME DESIGN	79
4.2.1 Game Design Document (GDD)	79
4.2.2 Technical Design Document (TDD).....	98

4.3 EVENTOS DO JOGO	105
4.4 SPI CITY.....	111
5 AVALIAÇÃO	125
5.1 PLANEJAMENTO DO EXPERIMENTO.....	125
5.1.1 Parte I – Definição do Experimento	125
5.1.2 Parte II – Planejamento do Experimento	127
5.1.3 Parte III – Operação do Experimento.....	131
5.1.4 Parte IV – Análise e Interpretação dos Dados.....	132
5.2 AVALIAÇÃO QUALITATIVA REALIZADA NA TURMA DE ESPECIALIZAÇÃO.....	133
5.3 EXPERIMENTO PRESENCIAL.....	133
5.3.1 Análise dos Resultados.....	134
5.4 EXPERIMENTO ONLINE.....	139
5.4.1 Análise dos resultados	140
5.5 RESULTADOS.....	141
5.5.1 Análise Quantitativa	141
5.5.2 Análise Qualitativa.....	143
5.5.3 Síntese e Análise dos Resultados	150
5.5.4 Ameaças à Validade	152
6 CONCLUSÕES	154
6.1 CONTRIBUIÇÕES DA PESQUISA	157
6.2 TRABALHOS FUTUROS	158
REFERÊNCIAS.....	160

1 INTRODUÇÃO

O mercado mundial de software e serviços é um dos que mais cresce nos últimos anos, chegando ao valor de 941 bilhões de dólares em 2011 segundo dados da ABES (Associação Brasileira de Empresas de Software) (ABES, 2011). No Brasil, neste mesmo ano, somente o mercado de software conseguiu alcançar um faturamento de 6,3 bilhões de dólares, e o de serviços 15,1 bilhões de dólares, representando os dois juntos 2,1% do mercado mundial.

Mas apesar desses números animadores, outras pesquisas mostram que as organizações de software continuam enfrentando problemas com prazo, custo, qualidade e satisfação dos seus clientes. Segundo pesquisa realizada pelo PMI (Project Management Institute) (PMI, 2010), 78% das empresas pesquisadas enfrentam problemas com cumprimento dos prazos estabelecidos para os projetos, 61% com custos, 56% com a qualidade de seus projetos e, 34% com a satisfação de seus clientes. Os problemas apresentados fazem com que as organizações tenham que buscar maneiras de melhorar a sua capacidade de desenvolvimento de software.

A melhoria de processo de software parte do pressuposto que é possível obter produtos de melhor qualidade a partir de melhorias na maneira como a organização desenvolve software, ou seja, a partir de melhorias no seu processo de desenvolvimento de software (HUMPHREY, 1987). A ISO/IEC 15504 (2003) define a melhoria de processo como a atividade de realizar modificações nos processos da organização para que eles estejam alinhados aos seus objetivos de negócio. Envolve o estabelecimento, a avaliação e a melhoria da capacidade dos processos mais importantes para organização, e é normalmente baseada em modelos de capacidade de processo de software (SALVIANO, 2006).

Modelos de capacidade de processo de software constituem um conjunto de boas práticas de Engenharia de Software, organizados de maneira sistemática, para guiar melhorias de processo de software (SALVIANO, 2006). São exemplos destes modelos o CMMI (Capability Maturity Model Integration) (SEI, 2010), o MPS.BR (Melhoria do Processo de Software Brasileiro) (SOFTEX, 2011a), e a ISO/IEC 15504 (2003). Há também a Norma ISO/IEC 12207 (2008), que apesar de não ser um modelo de capacidade de processo de software, é utilizada como referência por esses modelos.

Esses modelos de capacidade de processo usualmente integram um framework de modelos de referências e métodos de avaliação. O modelo de referência apresenta os requisitos que devem ser atendidos para as organizações estarem em conformidade com o modelo de capacidade de processo (SEI, 2010; SOFTEX, 2011a). O método de avaliação (SEI, 2006; SOFTEX, 2012) descreve o método que deve ser utilizado em uma avaliação oficial de um modelo de capacidade de processo. O objetivo dessas avaliações é verificar a aderência dos processos da organização com um modelo de capacidade de processo.

Mas apesar da importância da melhoria de processo, as organizações têm encontrado dificuldades para implementá-la. A maior dificuldade é encontrar pessoas especializadas para conduzir iniciativas de melhoria, fazendo com que as empresas tenham que contratar organizações de consultoria em melhoria de processo de software. O que dificulta ainda mais para micro, pequenas e médias empresas, em que os recursos financeiros são menores, a realizarem a melhoria (MONTONI, 2007).

Mesmo assim pesquisas têm mostrado que empresas que investem em melhoria de processo de software, têm obtido resultados positivos, como o aumento da satisfação dos seus clientes, precisão nas estimativas, aumento da produtividade, melhora na qualidade de seus produtos, e redução dos custos de seus projetos (SEI, 2007; TRAVASSOS & KALINOWSKI, 2012).

1.1 PROBLEMA DE PESQUISA

Muitas pesquisas têm sido feitas nos últimos anos visando identificar os fatores que afetam as iniciativas de melhoria de processo de software (ROCHA et al., 2005; NIAZI, WILLSON & ZOWGHI, 2006; ROCHA & WEBER, 2008; HABIB, 2009; NIAZI, WILLSON & ZOWGHI, 2009). Segundo essas pesquisas, um fator que é crítico para o sucesso das iniciativas de melhoria é a capacitação das pessoas envolvidas. Sem um treinamento apropriado nas áreas de processo chave para a organização, os programas de melhoria falham (HABIB, 2009).

A implementação de melhoria de processo de software é uma atividade fortemente dependente do conhecimento das pessoas envolvidas (MONTONI *et al.*, 2007). Os responsáveis pela implementação de melhoria nas organizações devem possuir um profundo conhecimento em Engenharia de Software e nos modelos de qualidade, para poder identificar de que forma uma prática já estabelecida na organização pode ser modificada ou como estabelecer novas práticas

inovadoras para produzir melhores resultados nos processos da organização (MONTONI *et al.*, 2007).

No entanto, pesquisas têm demonstrado que a principal dificuldade na implementação de melhoria de processo de software está justamente relacionada com a competência da equipe da empresa onde está sendo implementada a melhoria (ROCHA *et al.*, 2005; SOUZA & OLIVEIRA, 2005; FERNANDES *et al.*, 2007; HABIB, 2009; RODRIGUES & KIRNER, 2010). Destacam-se como as principais dificuldades a falta de conhecimento sobre as técnicas básicas da Engenharia de Software (ROCHA *et al.*, 2005) e a falta de entendimento nos modelos de capacidade de processo de software (TSUKUMO *et al.*, 2006). Estas dificuldades mostram a importância de se estabelecer estratégias adequadas de capacitação dos envolvidos em iniciativas de melhoria.

Tradicionalmente, o método utilizado para capacitação em iniciativas de melhoria são os treinamentos. Esses treinamentos normalmente ocorrem em duas fases, dependendo da estratégia de implementação adotada pela organização. Na fase de preparação do projeto de melhoria, são ministrados treinamentos de introdução aos modelos de capacidade de processo de software, com maior foco nos modelos que serão utilizados no projeto (TSUKUMO *et al.*, 2006). Nesse treinamento é apresentada uma visão geral sobre os processos abordados pelos modelos, para que as organizações possam identificar quais processos são mais relevantes de acordo com seus objetivos de negócio. Na fase de treinamento, são abordados os processos que serão implementados na organização e normalmente envolve a realização de um treinamento para cada processo com duração média de dezesseis horas (THIRY *et al.*, 2007). Nesses treinamentos, são abordadas as práticas e resultados esperados na execução de cada processo.

Além desses treinamentos, existem também os cursos oficiais dos modelos de capacidade de processo, como por exemplo: o curso de Introdução ao MPS.BR (SOFTEX, 2010a), que apresenta uma visão geral sobre o modelo de referência MR-MPS (SOFTEX, 2011a), sobre o método de avaliação MA-MPS (SOFTEX, 2012) e sobre o modelo de negócio MN-MPS (SOFTEX, 2010d), com duração de dezesseis horas. Existem também os cursos para os profissionais de melhoria de processo de software, como o curso para implementador (SOFTEX, 2010b) e o curso para avaliador do MPS.BR (SOFTEX, 2010c).

Esses cursos e treinamentos consistem basicamente de aulas expositivas, o que segundo Wangenheim, Hauck e Wangenheim (2009a), não atendem os objetos da melhoria, pois possuem o

enfoque muito teórico. Ou quando abrangem a prática dos processos, não se baseiam em exemplos concretos da organização, ou pelo menos do domínio de aplicação de cada organização específica que sejam familiares aos participantes (WANGENHEIM, HAUCK & WANGENHEIM, 2009a). E ainda segundo os mesmos autores, levando em consideração a complexidade que envolve a implementação de melhoria de processo de software, não se pode esperar que cursos e treinamentos predominantemente teóricos e de curta duração, sejam suficientes para capacitar as pessoas envolvidas na melhoria.

Segundo Santana (2007), algumas habilidades não serão desenvolvidas em cursos e treinamentos de natureza técnica, como por exemplo: reflexão e aprendizagem sobre a ação; noções de como o uso da teoria impactam em uma organização; empatia; capacidade de negociação; facilitação de grupos; e comunicação clara. Por outro lado, essas habilidades podem ser adquiridas e aprimoradas em ambientes que permitam liberdade para aprender através do fazer, um ambiente de risco relativamente baixo, e com instrutores que ajudem os estudantes a verem por eles mesmos e das suas maneiras, o que eles mais precisam saber (SANTANA, 2007).

Como consequência do treinamento inadequado grande parte do tempo das consultorias, que deveriam ser gastos com a elaboração do processo da organização, é gasto com a capacitação dos colaboradores (SOUZA & OLIVEIRA, 2005). Além disso, pessoas sem o conhecimento necessário, tendem a ser mais resistentes a introdução de novas práticas na organização, e terão maior dificuldade em reconhecer os benefícios da melhoria e em executar essas novas práticas (BARRETO *et al.*, 2006).

Para minimizar estes problemas uma estratégia que pode ser utilizada é a realização de *mentoring*¹ (ROCHA & WEBER, 2008). No entanto, o *mentoring* tem como consequência o aumento do custo do projeto de melhoria, pois requer mais tempo da equipe de consultoria (SOUZA & OLIVEIRA, 2005). Desta forma, nota-se a necessidade de estratégias inovadoras de capacitação, que aumentem o aprendizado dos participantes e não impacte no custo do projeto de melhoria.

¹ Envolve a transferência de conhecimento para os membros da organização durante a execução do processo (BARRETO *et al.*, 2006).

1.1.1 Solução Proposta

Observando a crescente adoção do modelo MPS.BR (TRAVASSOS & KALINOWSKI, 2012) e o resultado do mapeamento sistemático (Capítulo 3) que apontou que o ensino de melhoria de processo de software é pouco explorado pelos jogos educacionais existentes, é proposto o desenvolvimento de um jogo educacional para apoio ao ensino de melhoria de processo de software com foco no Nível G de maturidade do MPS.BR.

Apesar de já existir um jogo com objetivos similares ao jogo proposto (ADORNO, 2012) em que não se conseguiu obter resultados positivos na aprendizagem de melhoria de processo de software por meio da utilização do jogo, neste trabalho é proposto uma abordagem diferente na maneira como o conteúdo é abordado. No jogo desenvolvido no trabalho de Adorno (2012), assim como em outros jogos (SMITH & GOTEL, 2008; WANGENHEIM *et al.*, 2009; SILVA, 2010; GOLÇALVES, 2010), o conteúdo é explorado por meio de perguntas e respostas diretas sobre os assuntos abordados nesses jogos. No jogo proposto o conteúdo é explorado através da simulação de situações típicas de empresas de desenvolvimento de software, relacionadas com os resultados esperados do MPS.BR, porém não são realizadas perguntas diretas sobre o que deve ser realizado em cada resultado esperado. Para resolver essas situações o jogo simula uma conversa entre o jogador e os personagens do jogo, e somente após uma situação ser resolvida é que os resultados esperados associados com aquela situação são apresentados.

A proposta busca estabelecer um ambiente onde o jogador deve interagir com ele para evoluir no jogo. Além de situações típicas que ocorrem em empresas de software, o jogo permite também que o jogador possa realizar atividades como contratar e capacitar funcionários, interagir com o gerente de projetos, etc.

Em relação as suas características, o jogo proposto é um jogo digital com gráficos e animações em 2D, desenvolvido em Flash, executado via web, classificado como um jogo educativo (CRAWFORD, 1997) no estilo simulador (PRENSKY, 2007). A metáfora do jogo é a de um ambiente empresarial, tendo como cenário principal o ambiente de uma empresa de desenvolvimento de software.

Espera-se que o jogo proposto permita o entendimento e aplicação de conceitos de melhoria de processo de software. Neste sentido, as perguntas de pesquisa deste trabalho são:

1. A utilização do jogo proposto melhora o aprendizado de melhoria de processo de software?
2. A utilização do jogo proposto facilita o entendimento de conceitos de melhoria de processo de software?
3. A utilização do jogo proposto permite ao aluno criar um conhecimento em nível de aplicação de conceitos de melhoria de processo de software?
4. O jogo proposto é considerado uma atividade motivadora?

As perguntas de pesquisa deste trabalho estão concentradas na avaliação da efetividade da utilização de um jogo educacional para o ensino de melhoria de processo de software. Para responder essas perguntas foram elaboradas algumas hipóteses que se avaliadas podem fornecer indícios da corretude destas respostas. As hipóteses abaixo foram elaboradas com base nas perguntas de pesquisa apresentadas anteriormente e no objetivo geral do trabalho:

1. As pessoas que utilizam o jogo proposto apresentam melhora no aprendizado de melhoria de processo de software em relação as pessoas que não utilizam o jogo.
2. As pessoas que utilizam o jogo proposto conseguem uma melhor compreensão dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo.
3. As pessoas que utilizam o jogo proposto aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo.
4. As pessoas que utilizam o jogo consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software.

Na seção de metodologia serão discutidos os procedimentos para avaliar cada uma dessas hipóteses.

1.1.2 Delimitação de Escopo

O presente trabalho se limita na elaboração e desenvolvimento de um jogo para apoio ao ensino de melhoria de processo de software. O jogo será focado inicialmente no modelo de Melhoria de Processo de Software Brasileiro, o MPS.BR (SOFTEX, 2011a). Em relação aos níveis

de maturidade deste modelo, serão considerados somente os processos que compreendem o nível G de maturidade. E em relação aos níveis de capacidade, os diálogos do jogo estão mais voltados para o nível de capacidade 1, apesar do nível G de maturidade requerer o nível de capacidade 2, neste trabalho este nível de capacidade não foi explorado.

Embora outros modelos como, por exemplo, o CMMI, estejam fortemente ligados ao tema deste trabalho, os mesmos não serão objeto de estudo. Também não faz parte do escopo deste trabalho desenvolver novos conceitos ou técnicas de melhoria de processo de software, nem estabelecer novas ementas para cursos desta área.

O público alvo deste jogo é formado por profissionais da área de desenvolvimento de software que estejam participando de iniciativas de melhoria. É suposto que os usuários tenham conhecimento básico em Engenharia de Software.

O objetivo é desenvolver um jogo de computador, que possa ser utilizado como apoio ao ensino de melhoria de processo de software, com cenários, personagens, papéis, eventos, e interações, que permita o usuário aprender fazendo, ou seja, praticando melhoria de processo de software.

1.1.3 Justificativa

Este trabalho se justifica nas dificuldades apresentadas referente a capacitação em iniciativas de melhoria de processo de software. Como apresentado na Seção 1.1, a capacitação em iniciativas de melhoria não tem alcançado os objetivos pretendidos, pois tem se baseado em estratégias muito teóricas e pouco práticas (WANGENHEIM, HAUCK & WANGENHEIM, 2009a).

Neste sentido, espera-se que o uso de jogos educacionais possa contribuir para a capacitação em iniciativas de melhoria, pois eles permitem a simulação de cenários equivalentes ao do mundo real, com situações realísticas, que permitem as pessoas aprenderem fazendo, através de um ambiente de riscos relativamente baixos, aumentando a confiança para lidar com situações similares no mundo real (WANGENHEIM *et al.*, 2009). Além disso, o fato de jogos serem ferramentas de entretenimento tornam o aprendizado mais divertido, sendo assim, um forte fator motivacional (HOGLE, 1996; KAFAI, 2001; GIBSON, ALDRICH & PRENSKY, 2007).

Apesar de já existir o jogo 123SPI (ADORNO, 2012) que possui objetivos similares ao jogo proposto neste trabalho e que não obteve resultados positivos, esse trabalho propõe uma abordagem diferente em como o conteúdo é abordado no jogo.

Outros trabalhos similares foram identificados no contexto da Engenharia de Software (NAVARRO, 2006; BENITTI & MOLLÉRI, 2008; DANTAS, BARROS & WERNER, 2004; KIELING & ROSA, 2006; LUDEWIG, 2003; WANGENHEIM *et al.*, 2009). No entanto, considerando a aplicação de jogos para a melhoria de processo de software, não foram encontradas mais nenhuma iniciativa, além do 123SPI.

Por fim, no âmbito econômico e social, com a utilização do jogo espera-se melhorar a capacitação realizada em iniciativas de melhoria de processo de software, oferecendo uma ferramenta inovadora e de baixo custo, reduzindo assim os custos com capacitação, e aumentando as chances de sucesso dessas iniciativas. O impacto no custo existe, apesar de ser menor, pois o jogo permite aos participantes aprimorarem seus conhecimentos em melhoria de processo de software sem a necessidade de um instrutor, e é disponibilizado de forma gratuita.

1.2 OBJETIVOS

Esta seção apresenta os objetivos deste trabalho, conforme descrito a seguir.

1.2.1 Objetivo Geral

Desenvolver um jogo para apoiar o ensino de melhoria de processo de software.

1.2.2 Objetivos Específicos

1. Identificar uma metáfora que permita o envolvimento dos participantes com os cenários simulados.
2. Elaborar os conteúdos explorados no jogo incluindo design instrucional, materiais didáticos e mecanismos de avaliação.
3. Desenvolver o jogo para a plataforma web com aspectos de imersão, permitindo mecanismos de interação com os personagens e baseado em diálogos dinâmicos.

4. Avaliar o jogo a partir de um experimento que permitirá verificar as hipóteses levantadas.

1.3 METODOLOGIA

Nesta seção, a metodologia utilizada na pesquisa é classificada e uma síntese dos procedimentos metodológicos utilizados para o desenvolvimento da dissertação é apresentada. Neste sentido, esta seção está dividida nas subseções metodologia da pesquisa e procedimentos metodológicos, conforme apresentadas a seguir.

1.3.1 Metodologia da Pesquisa

Um método científico é o conjunto de processos ou operações mentais que devem ser empregados na investigação, permitindo que seja estabelecida uma linha de raciocínio para o processo de pesquisa (BARROS & LEHFELD, 2000). Neste trabalho, foi adotado o método dedutivo que tem como objetivo explicar o conteúdo de premissas a partir de uma sequência de raciocínio (BARROS & LEHFELD, 2000). O foco era estabelecer uma cadeia de raciocínio que permitisse confirmar as hipóteses de pesquisa apresentadas.

Do ponto de vista de sua natureza, esta pesquisa se caracteriza como uma pesquisa aplicada. A pesquisa aplicada tem por objetivo, gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos, envolvendo verdades e interesses locais (SILVA & MENEZES, 2001). Neste contexto, a construção do jogo permitiu avaliar a aplicabilidade desta ferramenta na formação de conhecimento em nível de aplicação de melhoria de processo de software, além de permitir que esta ferramenta possa ser utilizada como um complemento para os cursos desta área.

Do ponto de vista da forma de abordagem do problema, a pesquisa adotou tanto uma pesquisa qualitativa quanto uma pesquisa quantitativa. A pesquisa qualitativa considera que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números (SILVA, MENEZES, 2001). Neste caso, foi realizada uma avaliação subjetiva do jogo (focada na satisfação e percepção dos usuários), em que a interpretação dos fenômenos e a atribuição de significados foram realizadas pelo pesquisador. A pesquisa quantitativa considera que tudo pode ser quantificável, o que significa traduzir em números opiniões e informações, para classificá-las e analisá-las (SILVA &

MENEZES, 2001). Neste trabalho, foram coletadas medidas objetivas a partir da execução de um experimento, que permitiram a avaliação das hipóteses estabelecidas.

Do ponto de vista dos seus objetivos, esta pesquisa é essencialmente exploratória. A pesquisa exploratória visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses (SILVA & MENEZES, 2001). Além das hipóteses levantadas na Seção 1.1.1, foi utilizado o procedimento técnico da pesquisa bibliográfica para fundamentar e guiar o desenvolvimento da pesquisa.

1.3.2 Procedimentos Metodológicos

Este trabalho foi desenvolvido em seis etapas. Na primeira etapa foi feita a fundamentação teórica do trabalho, que abordou as áreas de melhoria de processo de software e jogos educacionais. Para esta etapa foi utilizado o procedimento técnico de pesquisa bibliográfica.

Na segunda etapa foi realizado um mapeamento sistemático, que visava identificar como o conteúdo dos modelos de capacidade de processo de software tem sido explorado pelos jogos educacionais existentes. Para esta etapa foi utilizado o procedimento técnico de mapeamento sistemático.

Na terceira etapa foi definida uma metáfora para o jogo. Para definição da metáfora foi feito uma pesquisa bibliográfica para identificar metáforas para jogos, além disso, foram realizadas reuniões entre o orientador e o mestrand, para o levantamento de idéias de metáforas. A metáfora definida foi a de um ambiente empresarial, por se tratar de um ambiente comum ao público alvo do jogo, permitindo aos jogadores associarem a situações vivenciadas no jogo com situações do seu dia a dia.

Na quarta etapa foram elaborados o roteiro do jogo e os diagramas da UML que descrevem como o jogo foi construído. O roteiro do jogo é composto pelo design instrucional, materiais didáticos e mecanismos de avaliação. Para elaboração do roteiro do jogo, foi utilizado o procedimento técnico de pesquisa bibliográfica para buscar por trabalhos correlatos para tomar como base.

Na quinta etapa o jogo foi desenvolvido de acordo com o projeto elaborado na etapa anterior, além disso, o jogo foi testado buscando identificar e solucionar erros.

Na sexta e última etapa o jogo foi avaliado por meio de experimentos buscando o levantamento de dados objetivos e subjetivos permitindo confirmar ou refutar as hipóteses de pesquisa. Neste experimento foram aplicados um teste antes de executar o jogo e um após a execução do jogo. Após a execução do jogo também foi aplicado um questionário qualitativo visando verificar a opinião dos participantes sobre o jogo. Com os resultados dos testes e do questionário qualitativo foi possível coletar os dados objetivos e subjetivos necessários para avaliação das hipóteses.

1.4 ESTRUTURA DA DISSERTAÇÃO

O presente trabalho está organizado em seis capítulos. O Capítulo 1, Introdução, apresentou por meio de sua contextualização o tema proposto nesse trabalho. Da mesma forma foram estabelecidos os resultados esperados por meio da definição de seus objetivos e apresentadas às limitações do trabalho permitindo uma visão clara do escopo proposto.

O Capítulo 2 apresenta a fundamentação teórica, sendo abordados os temas de melhoria de processo de software e jogos, respectivamente abordados nas seções 2.1 e 2.2. A fundamentação teórica sobre melhoria de processo de software contempla os conceitos fundamentais para a formulação do conteúdo instrucional do jogo proposto. Os conceitos jogos formam a base para o projeto e construção do jogo proposto.

O Capítulo 3 apresenta o mapeamento sistemático realizado com o objetivo de verificar como o conteúdo dos principais modelos de capacidade de software tem sido explorado pelos jogos educacionais existentes.

O Capítulo 4 apresenta o projeto do jogo para apoio ao ensino de melhoria de processo de software. Inicialmente são apresentados os objetivos educacionais do jogo. Em seguida é apresentado o game design, no qual são apresentados dois documentos conhecidos como *Game Design Document* (GDD) e *Technical Design Document* (TDD). Após o game design são apresentados os eventos do jogo, que são as situações que o jogador terá que resolver no jogo. Por fim, é apresentada uma simulação da execução do jogo, tomando como base as telas do mesmo.

O Capítulo 5 apresenta o planejamento, aplicação e os resultados das avaliações realizadas.

Por fim no Capítulo 6 são apresentadas as conclusões do trabalho, as contribuições da pesquisa e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo visa apresentar os conceitos e terminologias relacionadas com este trabalho. Primeiramente é abordada a área de melhoria de processo de software, com apresentação de definições de conceitos chaves, e dos principais modelos de capacidade de processo de software. Em seguida é abordada a área de jogos, sendo apresentadas a definição, características e classificações de jogos, a sua utilização com fins pedagógicos, como os jogos podem ser projetos com fins pedagógicos, e por fim, são apresentadas duas abordagens para avaliação de jogos educacionais.

2.1 MELHORIA DE PROCESSO DE SOFTWARE

A melhoria de processo de software parte do pressuposto que é possível obter produtos de melhor qualidade a partir de melhorias na maneira como a organização desenvolve software, ou seja, a partir de melhorias no seu processo de desenvolvimento de software (HUMPHREY, 1987). A ISO/IEC 15504 (2003) define a melhoria de processo como a atividade de realizar modificações nos processos da organização para que eles estejam alinhados aos seus objetivos de negócio. Envolve o estabelecimento, a avaliação e a melhoria da capacidade dos processos mais importantes para organização, e é normalmente baseada em modelos de capacidade de processo de software (SALVIANO, 2006). Visando um melhor entendimento da área de melhoria de processo de software, a seguir são apresentadas algumas definições de conceitos chaves da área.

2.1.1 Definições

Esta seção apresenta as definições dos conceitos chaves de melhoria de processo de software.

2.1.1.1 Processo de Software

O Instituto de Engenharia de Software (*Software Engineering Institute - SEI*) define processo como um conjunto de atividades inter-relacionadas, que transforma entradas em saídas, para alcançar um determinado fim (SEI, 2010). Da mesma forma a ISO 9000 define processo como um conjunto de atividades inter-relacionadas ou interativas que transforma insumos (entradas) em produtos (saídas) (ISO 9000, 2005).

Processo de software pode ser definido como “um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e os produtos de trabalho associados (por exemplo, planos de projeto, documentos de projeto, código, casos de teste, e manuais de usuário)” (PAULK *et al.*, 1993). Processo de software é o que as pessoas fazem para um determinado propósito, utilizando suas habilidades e conhecimento, com o apoio de artefatos, ferramentas e outros recursos, para produzir software e seus produtos associados (SALVIANO, 2006).

Os termos “processo de software” e “processo de desenvolvimento de software” costumam ser utilizados como sinônimos, porém os mesmos são bem distintos (WEBER, 2005). O “processo de desenvolvimento de software” se restringe as atividades diretamente relacionadas ao desenvolvimento do produto de software, como por exemplo, a especificação, o projeto e a implementação do software, já o “processo de software” é mais abrangente, englobando inclusive atividades de apoio e os processos organizacionais envolvidos (HAUCK, 2007).

2.1.1.2 Capacidade de processo

A ISO/IEC 15504-1 define capacidade de processo como “uma caracterização da habilidade de um processo em atender objetivos requeridos ou contribuir (juntamente com outros processos) para o atendimento de objetivos requeridos” (ISO/IEC 15504-1, 2004, p. 4). O grau de refinamento e institucionalização com que um processo é executado numa organização é representado pela caracterização da capacidade deste processo (SOFTEX, 2011a). A capacidade de processo pode ser representada por níveis que estabelecem o grau de evolução dos processos, sendo que quanto mais alto for o nível de capacidade de um processo, mais organizado e institucionalizado se espera que ele seja (ZOUCAS, 2010). A ISO/IEC 15504 (2003) define seis níveis de capacidade de processo, que são:

- **Incompleto ou indefinido:** o processo não está implementado ou não consegue alcançar o seu propósito (ISO/IEC 15504-2, 2003);
- **Executado:** o processo alcança o seu propósito (ISO/IEC 15504-2, 2003). Um processo executado é um processo que realiza o trabalho necessário para produzir os seus produtos de trabalho. Mas apesar do processo executado trazer melhorias importantes, essas melhorias podem ser perdidas se elas não forem institucionalizadas (SEI, 2010);

- **Gerenciado:** o processo é implementado de maneira gerenciada (planejado, monitorado e melhorado) e seus produtos de trabalho são apropriadamente estabelecidos, controlados e mantidos (ISO/IEC 15504-2, 2003). Um processo gerenciado é um processo que: é executado e gerenciado de acordo com a política organizacional; emprega pessoas qualificadas com recursos adequados para produzir saídas controladas; envolve as partes interessadas; é monitorado, controlado e revisto; e é avaliado para verificar a aderência com sua descrição (SEI, 2010);
- **Estabelecido ou definido:** o processo gerenciado é agora implementado usando um processo definido e é capaz de alcançar os resultados do processo definido (ISO/IEC 15504-2, 2003). Um processo definido é um processo gerenciado que: é adaptado do conjunto de processos padrões da organização de acordo com as diretrizes de adaptação da organização; tem uma descrição do processo mantida; e contribui com experiências relacionadas ao processo com os ativos de processos organizacionais (SEI, 2010);
- **Previsível ou gerenciado quantitativamente:** o processo estabelecido ou definido agora opera dentro de limites estabelecidos para alcançar os resultados do processo, além disso, sua implementação é suportada e conduzida através de informações quantitativas derivadas de medições relevantes. O desempenho dos processos são gerenciados quantitativamente e se comportam de maneiras previsíveis para apoiar os objetivos de negócio da organização, e causas de variação podem ser tratadas (ISO/IEC 15504-2, 2003); e
- **Em otimização ou melhoria contínua:** o processo previsível é continuamente melhorado de forma ordenada e intencional para satisfazer os objetivos de negócio atuais e os planejados da organização. Este nível de capacidade depende fundamentalmente da compreensão quantitativa do comportamento do processo (ISO/IEC 15504-2, 2003).

2.1.1.3 Maturidade e Perfil de Capacidade

A maturidade de uma organização é caracterizada pela avaliação da capacidade de um conjunto selecionado de processos em relação a um perfil alvo de capacidade de processo (SALVIANO, 2006). Um perfil de capacidade pode ser entendido como um subconjunto dos processos definidos por um modelo ou norma, acompanhados do respectivo nível de capacidade desejado para cada processo deste subconjunto (SALVIANO, 2006). Este subconjunto pode

representar, inclusive, todos os processos do modelo ou norma considerado. Deste modo, o nível de maturidade da organização representará então um grau de melhoria de processo quando o perfil for alcançado. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos (SOFTEX, 2011a).

2.1.1.4 Representação Contínua e Estagiada

A representação por estágios ou estagiada está associada com alcance de níveis de maturidade, que são perfis de capacidade pré-fixados por um modelo ou norma. Se uma organização deseja alcançar um determinado nível de maturidade ela deve implementar todos os processos determinados pelo modelo ou norma para o nível desejado, além dos processos dos níveis anteriores ao desejado. A representação contínua está associada com o alcance de níveis de capacidade, não existe uma definição fixa do conjunto de processos que devem ser implementados, sendo assim, é a própria organização que estabelece o perfil de capacidade que ela irá implementar de acordo com os seus objetivos de negócio (SEI, 2010).

2.1.2 Modelos de capacidade de processo

Modelos de capacidade de processo constituem um conjunto de boas práticas de Engenharia de Software, organizados de maneira sistemática, para guiar melhorias de processo de software (SALVIANO, 2006). A seguir são apresentados os modelos e norma utilizadas como base neste trabalho.

2.1.2.1 CMMI

O CMMI - *Capability Maturity Model Integration* (Integração de Modelos de Maturidade da Capacidade) (SEI, 2010) é um framework para a melhoria de processos, composto pelas melhores práticas, que cobre todo o ciclo de vida de um produto, desde a sua concepção até sua entrega e a posterior manutenção. Este framework é desenvolvido pelo *Software Engineering Institute* (SEI) com a participação da indústria e do governo. Atualmente, o CMMI encontra-se na versão 1.3, e possui modelos para três áreas de interesses diferentes, que são: desenvolvimento, serviços e aquisição. O CMMI-DEV é o modelo para área de desenvolvimento de sistemas e possui os dois tipos de representação para melhoria de processo, contínua e estagiada.

Para dar suporte as organizações que utilizam a representação contínua, o modelo CMMI-DEV estabelece quatro níveis de capacidade. Para alcançar um nível de capacidade, todos os objetivos genéricos até o nível desejado devem ser satisfeitos. Os quatro níveis de capacidade do modelo CMMI-DEV são apresentados a seguir (SEI, 2010):

- **0 – Incompleto:** é um processo que não é executado ou é parcialmente executado. Neste nível um ou mais objetivos específicos da área de processo não são satisfeitos, e não existem objetivos genéricos para este nível;
- **1 – Executado:** é um processo que realiza o trabalho necessário para produzir os seus produtos de trabalho. Para alcançar este nível todos os objetivos específicos da área de processo devem ser satisfeitos, assim como o objetivo genérico 1;
- **2 – Gerenciado:** é um processo executado que: é planejado e executado de acordo com políticas; emprega pessoas qualificadas com recursos adequados para produzir saídas controladas; envolve as partes interessadas; é monitorado, controlado e revisto; e é avaliado para verificar a aderência com sua descrição. Para alcançar este nível todos os objetivos específicos da área de processo devem ser atendidos, assim como os objetivos genéricos 1 e 2; e
- **3 – Definido:** é um processo gerenciado que: é adaptado do conjunto de processos padrões da organização de acordo com as diretrizes de adaptação da organização; tem uma descrição do processo mantida; e contribui com experiências relacionadas ao processo com os ativos de processos organizacionais. Para alcançar este nível todos os objetivos específicos da área de processo devem ser atendidos, assim como os objetivos genéricos 1, 2 e 3.

Depois que uma organização conseguir atingir o nível 3 de capacidade nas áreas que ela selecionou para a melhoria, ela pode continuar sua jornada de melhoria tratando de áreas de processo de alta maturidade. Estas áreas de processo focam na melhoria do desempenho dos processos que já foram implementados. Elas utilizam estatística e outras técnicas quantitativas para melhorar os processos organizacionais e de projeto para melhor atender aos objetivos de negócio da organização (SEI, 2010).

Os níveis de maturidade no CMMI consistem de práticas específicas e genéricas relacionadas a um conjunto de áreas de processo que devem ser satisfeitas para que organização

alcance um determinado nível de maturidade. Cada nível de maturidade amadurece um subconjunto de processos da organização, preparando-os para o nível de maturidade seguinte. Os níveis de maturidade são medidos pelo alcance dos objetivos específicos e genéricos associados a cada conjunto predefinido de áreas de processo. O CMMI-DEV estabelece cinco níveis de maturidade, que são (SEI, 2010):

- **1 – Inicial:** Neste nível de maturidade os processos são executados de maneira informal e caótica. O sucesso depende da competência das pessoas e não do uso de processos comprovados;
- **2 – Gerenciado:** Neste nível de maturidade os requisitos, processos, produtos de trabalho e serviços são gerenciados. Os projetos são executados e gerenciados de acordo com seus planos. O status dos produtos de trabalho e tarefas é visível e compromissos são estabelecidos entre os *stakeholders* relevantes;
- **3 – Definido:** Neste nível de maturidade os processos são bem caracterizados e entendidos e estão descritos em padrões, procedimentos, ferramentas e métodos. Os projetos estabelecem seus processos definidos adaptando-os do conjunto de processos padrão da organização;
- **4 – Gerenciado quantitativamente:** Neste nível de maturidade o foco está em compreender e controlar o desempenho dos processos utilizando os resultados para gerenciar os projetos. Para isso, medidas específicas de desempenho são coletadas e analisadas utilizando estatística ou outros métodos quantitativos; e
- **5 – Em otimização:** Neste nível de maturidade o foco está em continuamente melhorar os processos, com base no entendimento das causas comuns de variação inerentes ao processo.

Os níveis de maturidade 2 e 3 usam intencionalmente os mesmos nomes dos níveis 2 e 3 de capacidade, por que os conceitos de capacidade e maturidade são complementares. Os níveis de maturidade são usados para caracterizar a melhoria organizacional relativo a um conjunto de áreas de processos, e os níveis de capacidade caracterizam a melhoria organizacional relativo a uma área de processo individual.

A estrutura do CMMI-DEV (ver Figura 1) é composta por alguns elementos básicos que devem ser estudados para que o modelo possa ser melhor entendido. Alguns desses elementos são apresentados a seguir (SEI, 2010):

- **Área de processo:** é um grupo de práticas relacionadas a uma área que, quando executadas de forma coletiva, satisfazem um conjunto de metas consideradas importantes para trazer uma melhoria significativa naquela área;
- **Objetivos específicos:** são objetivos que se aplicam a uma área de processo e tratam de características únicas que descrevem o que deve ser implementado para satisfazer a área de processo;
- **Objetivos genéricos:** são objetivos que se aplicam a várias áreas de processo e por isso são chamados de genéricos. Os objetivos genéricos descrevem as características que devem estar presentes para institucionalizar os processos que implementam as áreas de processo;
- **Práticas específicas:** são atividades consideradas importantes na satisfação de um objetivo específico associado. As práticas específicas descrevem as atividades que se espera resultarem no atendimento dos objetivos específicos de uma área de processo;
- **Práticas genéricas:** são atividades, associadas a um objetivo genérico, que são consideradas importantes para o alcance deste objetivo e contribuem para a institucionalização dos processos associados com uma área de processo;
- **Produtos de trabalho típicos:** é um componente informativo do modelo que apresenta exemplos de saídas para as práticas específicas ou genéricas; e
- **Sub-práticas:** é uma descrição detalhada que fornece orientações para a interpretação e implementação de uma prática específica ou genérica.

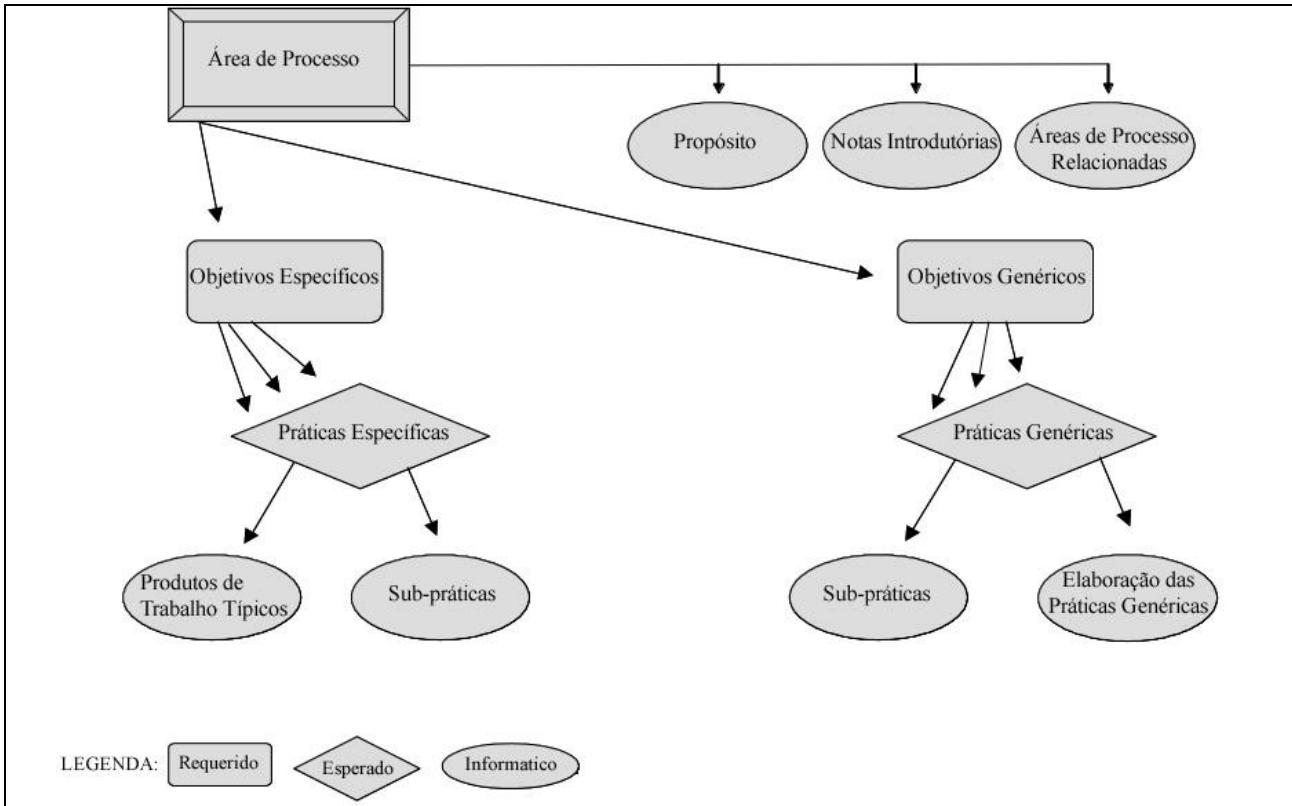


Figura 1. Estrutura básica do CMMI-DEV versão 1.3

Fonte: Adaptado de SEI (2010)

As 22 áreas de processo do modelo CMMI-DEV versão 1.3 e suas categorias e níveis de maturidade associados, são apresentadas na Tabela 1. Nesta tabela as áreas de processo estão identificadas por uma tradução do nome para o português e por uma sigla (baseada no nome em inglês).

Tabela 1. Áreas de processo, categorias e níveis de maturidade do CMMI-DEV

Área de Processo	Categoria	Nível de Maturidade
Análise Casual e Resolução (CAR)	Supporte	5
Gerência de Configuração (CM)	Supporte	2
Análise de Decisão e Resolução (DAR)	Suporte	3
Gerência de Projetos Integrada (IPM)	Gerência de Projetos	3
Medição e Análise (MA)	Suporte	2
Definição de Processo Organizacional (OPD)	Gerência de Processo	3
Foco no Processo Organizacional (OPF)	Gerência de Processo	3
Gerência de Performance Organizacional (OPM)	Gerência de Processo	5
Performance de Processo Organizacional (OPP)	Gerência de Processo	4
Treinamento Organizacional (OT)	Gerência de Processo	3
Integração de Produto (PI)	Engenharia	3
Monitoração e Controle de Projeto (PMC)	Gerência de Projetos	2
Planejamento de Projetos (PP)	Gerência de Projetos	2
Garantia da Qualidade de Processo e Produto (PPQA)	Suporte	2
Gerência de Projeto Quantitativa (QPM)	Gerência de Projetos	4
Desenvolvimento de Requisitos (RD)	Engenharia	3
Gerência de Requisitos (REQM)	Gerência de Projetos	2
Gerencia de Riscos (RSKM)	Gerência de Projetos	3
Gerência de Acordo com Fornecedor (SAM)	Gerência de Projetos	2
Solução Técnica (TS)	Engenharia	3
Validação (VAL)	Engenharia	3
Verificação (VER)	Engenharia	3

2.1.2.2 ISO/IEC 15504

A ISO/IEC 15504 é uma norma internacional para avaliação de processos, elaborada pela ISO (*International Organization for Standardization*) e pela IEC (*International Electrotechnical Commission*), com apoio do projeto SPICE (*Software Process Improvement and Capability dEtermination*) (EL EMAM, DROUIN & MELO, 1998; SALVIANO, 2006; ISO/IEC 15504, 2003). O desenvolvimento da ISO/IEC 15504 começou em 1993 a partir da constatação da necessidade de uma norma internacional de avaliação de processo de software. Foi criado então o projeto SPICE com uma equipe de especialistas internacionais para apoiar o desenvolvimento das versões iniciais da futura norma e coordenar a utilização destas versões pela comunidade (SALVIANO, 2006). Esta norma provê uma abordagem estruturada para a avaliação de processos com os seguintes objetivos (ISO/IEC 15504, 2003):

- Compreender a situação dos processos de uma organização visando a sua melhoria;
- Determinar a adequação dos processos de uma organização para um requisito específico ou uma classe de requisitos; e

- Determinar a adequação dos processos de um fornecedor para um contrato específico ou uma classe de contratos.

A abordagem para avaliação de processos da ISO/IEC 15504 foi projetada para prover uma base comum para descrever os resultados da avaliação de processo, permitindo algum grau de comparação da avaliação baseada em diferentes, mas compatíveis, modelos e métodos. A Figura 2 mostra os principais elementos e relacionamentos do modelo conceitual da ISO/IEC 15504.

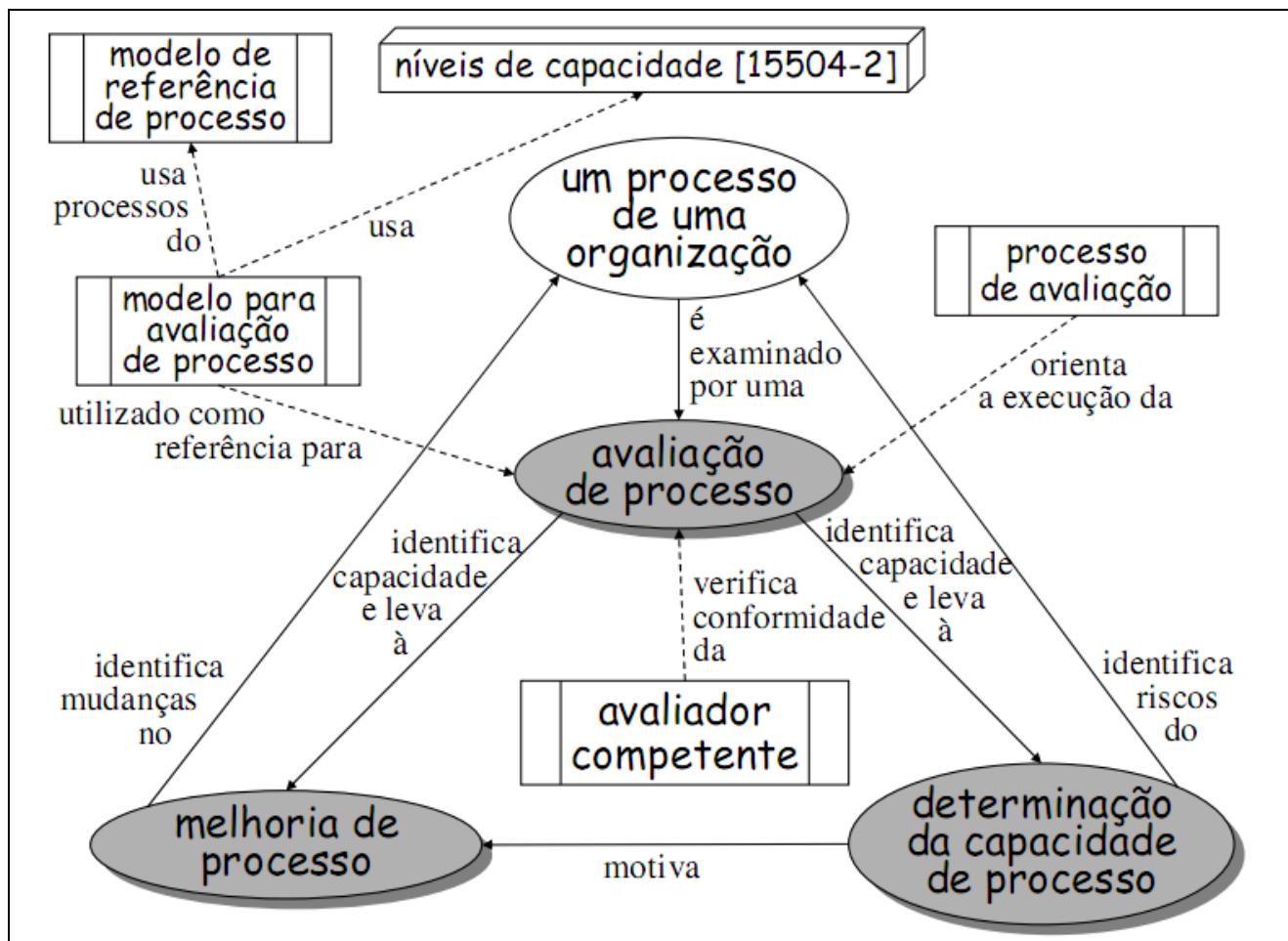


Figura 2. Modelo conceitual da ISO/IEC 15504

Fonte: Salviano (2006)

Como pode ser observado na Figura 2, uma avaliação de processo examina de forma disciplinada um processo em relação a um modelo para avaliação de processo, é realizada seguindo a orientação de um processo de avaliação, e sua conformidade é verificada por um avaliador competente, utilizando os níveis de capacidade definidos na ISO/IEC 15504-2 e processos de um ou mais modelos de referência de processo (SALVIANO, 2006).

A versão de 2004 da ISO/IEC 15504 é composta por cinco partes, sendo que destas só a parte 2 é normativa. A seguir é apresentada uma breve descrição do conteúdo de cada uma dessas partes (ISO/IEC 15504-1, 2004):

- **Parte 1 – Conceitos e vocabulário:** é uma parte informativa que: descreve como as partes se encaixam e oferece orientações para a sua seleção e uso; estabelece os requisitos exigidos pela ISO/IEC 15504 e sua aplicabilidade na realização de uma avaliação; e contém um glossário de termos e definições relacionadas à avaliação;
- **Parte 2 – Realização de uma avaliação:** é uma parte normativa que: define os requisitos normativos para a avaliação do processo e para modelos de processo de uma avaliação; e define o *framework* de medição para avaliar a capacidade do processo. O *framework* de medição define nove atributos de processo que são agrupados em seis níveis de capacidade de processo;
- **Parte 3 – Orientações para realização de uma avaliação:** é uma parte informativa que apresenta orientações para atender aos requisitos para execução de uma avaliação definidos na parte 2;
- **Parte 4 – Orientações de uso para aprimoramento e determinação de capacidade do processo:** é uma parte informativa que traz orientações para avaliação de processo com objetivos de aprimoramento do processo e determinação da capacidade; e
- **Parte 5 – Exemplo de um modelo de avaliação:** é uma parte informativa que apresenta um modelo para realizar avaliação de processo, baseado e diretamente compatível com o modelo de referência de processo da ISO/IEC 12207 (2008).

A avaliação de processo na norma ISO/IEC 15504 é fundamentada em um modelo bidimensional, contendo uma dimensão de processo e uma dimensão de capacidade. A dimensão de processo é fornecida por um modelo de referência de processo externo, que define um conjunto de processos caracterizados por declarações contendo o seu propósito e resultados. A dimensão de capacidade do processo consiste na estrutura de medição que compreende seis níveis de capacidade de processo e seus atributos de processo associados (ISO/IEC 15504-2, 2003). A Tabela 2 apresenta os níveis de capacidade e os atributos de processo da norma ISO/IEC 15504.

Tabela 2. Atributos de processo e níveis de capacidade da ISO/IEC 15504

Níveis de Capacidade	Atributos de Processo
Nível 0: Processo incompleto	
Nível 1: Processo executado	PA 1.1: Atributo de execução do processo
Nível 2: Processo gerenciado	PA 2.1: Atributo de gerência de execução PA 2.2: Atributo de gerência de produto de trabalho
Nível 3: Processo estabelecido	PA 3.1: Atributo de definição de processo PA 3.2: Atributo de implementação de processo
Nível 4: Processo previsível	PA 4.1: Atributo de medição de processo PA 4.2: Atributo de controle de processo
Nível 5: Processo em otimização	PA 5.1: Atributo de inovação de processo PA 5.2: Atributo de otimização de processo

Fonte: ISO/IEC 15504-2 (2003)

A medição da capacidade é baseada em um conjunto de atributos de processo (*process attribute* - PA), onde cada atributo define um aspecto particular da capacidade de processo (ver Tabela 2). A extensão do alcance do atributo de processo é medida utilizando uma escala ordinal de medição, como apresentado a seguir (ISO/IEC 15504-2, 2003):

- **N – Não atingido:** existe pouca ou nenhuma evidência do alcance do atributo no processo avaliado;
- **P – Parcialmente atingido:** existe alguma evidência de aproximação e algum alcance do atributo no processo avaliado;
- **L – Largamente atingido:** existe evidência de aproximação sistemática e de alcance significativo do atributo, porém podem existir alguns pontos fracos relacionados a este atributo no processo avaliado; e
- **F – Completamente atingido:** existe evidência de uma aproximação completa e sistemática e de alcance total do atributo, e não existem pontos fracos relevantes relacionados com este atributo no processo avaliado.

A parte 5 da norma ISO/IEC 15504 estabelece um modelo de avaliação de processo que pode ser utilizado quando uma avaliação for realizada de acordo com as especificações da ISO/IEC 15504-2. Este modelo de avaliação foi estabelecido com base no modelo de referência de processo da ISO/IEC 12207 e nos atributos de processo definidos na ISO/IEC 15504-2. O modelo de avaliação de processo compreende um conjunto de indicadores de execução e capacidade do

processo, que são utilizados como base para a coleta de evidências objetivas que permite um avaliador fazer classificações (ISO/IEC 15504-5, 2006).

O modelo de avaliação de processo é composto por processos agrupados em três categorias, como na ISO/IEC 12207, que são: processos fundamentais do ciclo de vida, processos de apoio ao ciclo de vida e processos organizacionais do ciclo de vida. Dentro de uma categoria, os processos são agrupados em um segundo nível, de acordo com o tipo de atividade abordada. Esses grupos são definidos a fim de auxiliar os avaliadores na definição do escopo da avaliação em termos de seleção de processo. A Figura 3 apresenta as categorias, os grupos e os processos da ISO/IEC 15504.

O modelo de avaliação de processo baseia-se no princípio de que a capacidade de um processo pode ser avaliada através da demonstração da realização dos atributos com base em evidências relacionadas aos indicadores de avaliação. Há dois tipos de indicadores de avaliação: indicadores de capacidade do processo, os quais se aplicam aos níveis de capacidade 1 a 5, e os indicadores de execução do processo, os quais se aplicam exclusivamente ao nível de capacidade 1. Os indicadores de capacidade de processo são (ISO/IEC 15505-5, 2006):

- **Prática Genérica (PG):** são atividades de um tipo comum e fornecem orientação sobre a implementação das características do atributo;
- **Recurso Genérico (RG):** são recursos relacionados que podem ser usados durante a execução do processo, a fim de alcançar o atributo. Esses recursos podem incluir recursos humanos, ferramentas, métodos e infra-estrutura; e
- **Produto do Trabalho Genérico (PTG):** são conjuntos de características que se espera que sejam evidentes em produtos de trabalho de tipos genéricos como resultado da realização de um atributo. Eles representam tipos básicos de produtos de trabalho que podem ser matérias para todos os tipos de processo ou produtos.



Figura 3. Categorias, Grupos de Processo e Processos da ISO/IEC 15504-5

Fonte: ISO/IEC 15504-5 (2006)

Os indicadores de execução do processo relacionam-se com processos individuais definidos na dimensão do processo do modelo de avaliação e são escolhidos a fim de abordar explicitamente o cumprimento do objetivo definido para o processo. Os dois tipos de indicadores de execução do processo são (ISO/IEC 15505-5, 2006):

- **Prática- Base (PB):** é uma atividade que trata do objetivo de um processo específico e cuja execução associada a um processo de maneira consistente ajudará o alcance de seu objetivo; e
- **Produtos de Trabalho (PT):** são resultados da execução do processo que podem ser identificados e úteis para a satisfação do objetivo do processo.

O modelo de avaliação de processo engloba todas as especificações definidas pela ISO/IEC 15504-2, no entanto, o uso deste modelo não é necessário para que as especificações desta norma sejam cumpridas, todos os outros modelos de avaliação que satisfaçam os requisitos da ISO/IEC 15504-2 podem ser utilizados em uma avaliação consistente desta norma (ISO/IEC 15504-5, 2006).

2.1.2.3 MPS.BR

O MPS.BR (Melhoria de Processo do Software Brasileiro), vem sendo desenvolvido desde dezembro de 2003, por sete conceituadas organizações nacionais (SOFTEX, 2011a). O MPS.BR está sendo desenvolvido visando principalmente auxiliar micro, pequenas e médias empresas de software brasileiras a adequar à sua realidade o correspondente aos níveis de maturidade 2 e 3 de modelos como o CMMI e a ISO/IEC 15504-5. O MPS.BR está dividido em três componentes principais (ver Figura 4), que são (SOFTEX, 2011a):

- **MR-MPS - Modelo de Referência:** contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS, contém as definições dos níveis de maturidade, processos e atributos do processo, e está em conformidade com os requisitos de modelos de referência da norma ISO/IEC 15504-2;
- **MA-MPS - Método de Avaliação:** contém o processo e o método oficial de avaliação formal do MR-MPS, e está em conformidade com os requisitos de modelos de avaliação da norma ISO/IEC 15504-2; e

- **MN-MPS - Modelo de Negócio:** descreve as regras de negócio para implementação do MR-MPS pelas Instituições Implementadoras (II), avaliação seguindo o MA-MPS pelas Instituições Avaliadoras (IA), organização de grupos de empresas pelas Instituições Organizadoras de Grupos de Empresas (IOGE) para implementação e avaliação do MR-MPS, certificação de Consultores de Aquisição (CA), e programas anuais de treinamento do MPS.BR por meio de cursos, provas e *workshops*.

Além desses componentes, o MPS.BR é composto por quatro guias (ver Figura 4), que são (SOFTEX, 2011a):

- **Guia Geral:** contém a descrição geral do MPS.BR e detalha o MR-MPS, seus componentes e as definições comuns necessárias para seu entendimento e aplicação;
- **Guia de Aquisição:** descreve um processo de aquisição de software e serviços correlatos, de forma a apoiar as instituições que queiram adquirir produtos de software e serviços correlatos apoiando-se no MR-MPS;
- **Guia de Avaliação:** descreve o MA-MPS, os requisitos para avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA); e
- **Guia de Implementação:** seria de dez documentos que fornecem orientações para implementar nas organizações os níveis de maturidade descritos no MR-MPS.

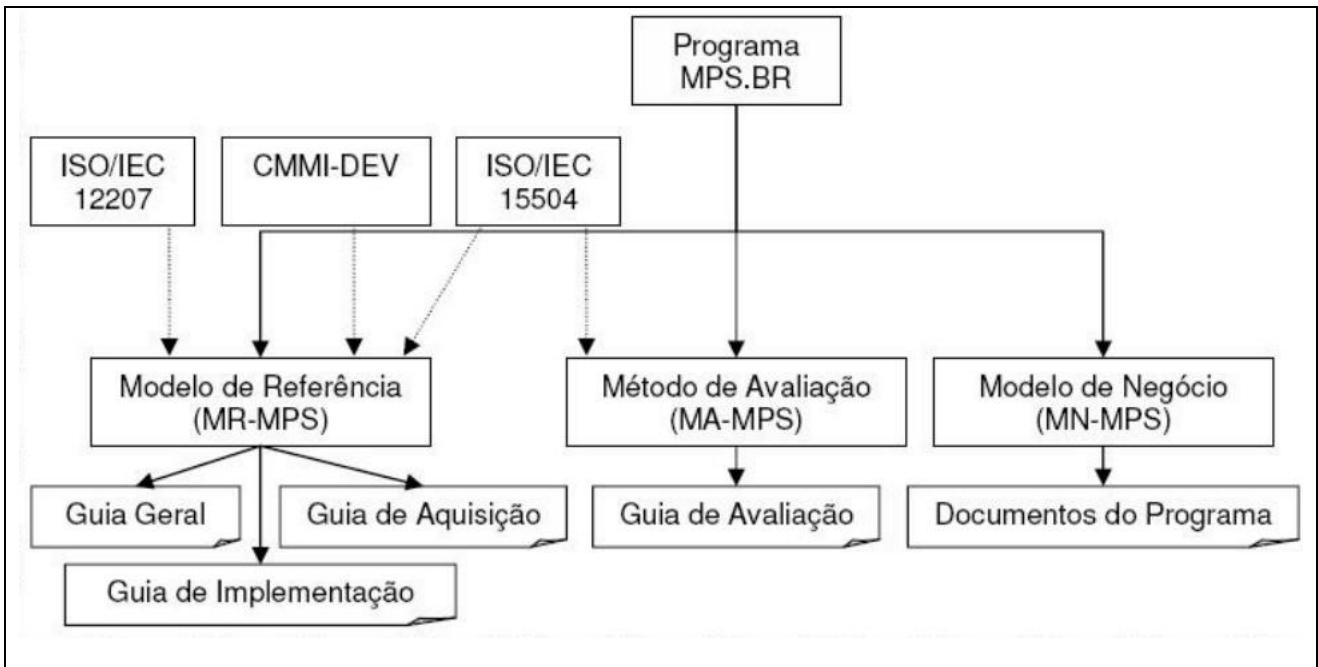


Figura 4. Componentes do MPS.BR

Fonte: SOFTEX (2011a)

O MR-MPS se baseia nos conceitos de maturidade e capacidade de processo para melhoria da qualidade dos produtos de software. Onde são definidos sete níveis de maturidade (SOFTEX, 2011a): A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). Sendo que o nível mais baixo desta escala é o nível G e o nível mais alto é o A. E nove atributos de processo (AP) (SOFTEX, 2009a): AP 1.1 – O Processo é executado; AP 2.1 – O processo é gerenciado; AP 2.2 – Os produtos de trabalho do processo são gerenciados; AP 3.1 – O processo é definido; AP 3.2 – O processo está implementado; AP 4.1 – O processo é medido; AP 4.2 – O processo é controlado; AP 5.1 – O processo é objeto de melhorias e inovações; e AP 5.2 – O processo é otimizado continuamente. Conforme a organização evoluí nos níveis de maturidade, um maior nível de capacidade para desempenhar o processo deve ser atingido pela organização. A Tabela 3 apresenta os níveis de maturidade do MR-MPS, os processos e os atributos de processo correspondentes a cada nível, onde os atributos de processo AP 4.1, AP 4.2, AP 5.1 e AP 5.2 devem ser implementados somente para os processos críticos da organização/unidade organizacional, selecionados para análise de desempenho. Os demais atributos de processo devem ser implementados para todos os processos (SOFTEX, 2011a).

Tabela 3. Níveis de maturidade do MR-MPS

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Desenvolvimento para Reutilização – DRU	
	Gerência de Decisões – GDE	
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
	Desenvolvimento de Requisitos – DRE	
E	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
	Avaliação e Melhoria do Processo Organizacional – AMP	
F	Medição – MED	AP 1.1, AP 2.1 e AP 2.2
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
	Aquisição – AQU	
G	Gerência de Requisitos – GRE	AP 1.1 e AP 2.1
	Gerência de Projetos – GPR	

Fonte: SOFTEX (2011a)

Os processos no MR-MPS são descritos em termos de propósito e resultados esperados, onde o propósito descreve o objetivo geral a ser alcançado durante a execução do processo e os resultados esperados estabelecem os resultados a serem obtidos com a efetiva implementação do processo. Estes resultados esperados podem ser evidenciados por um produto de trabalho ou por uma mudança significativa de estado ao ser executado o processo (SOFTEX, 2011a).

A capacidade do processo no MR-MPS é descrita por um conjunto de atributos de processo descritos em termos de resultados esperados (ver Tabela 3). O atendimento aos atributos de processo (AP) é alcançado pelo atendimento dos resultados esperados dos atributos de processo (RAP), e é requerido para todos os processos no nível correspondente ao nível de maturidade, embora eles não estejam detalhados dentro de cada processo. Ao passar para um nível de maturidade superior, os processos anteriormente implementados devem passar a ser executados no nível de capacidade exigido neste nível superior (SOFTEX, 2011a).

Como o foco do presente trabalho está no nível G, ele será descrito detalhadamente a seguir. O nível G é o primeiro nível de maturidade do MR-MPS, e é composto pelos processos Gerência de Projetos e Gerência de Requisitos. Neste nível a implementação dos processos devem satisfazer os atributos de processo AP 1.1 e AP 2.1 (SOFTEX, 2011a). A seguir os processos que compõem o nível de maturidade G são apresentados detalhadamente, como o foco deste trabalho está no nível de capacidade 1, os atributos de processo AP 1.1 e AP 2.1 não serão detalhados.

Gerência de Projetos (GPR)

- Propósito:
 - Estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade.
- Resultados esperados:
 - GPR1 - O escopo do trabalho para o projeto é definido
 - Este resultado espera que a organização defina todo o trabalho necessário, e somente ele, para entregar um produto que satisfaça as necessidades, características e funções especificadas para o projeto, de forma a concluir-lo com sucesso.
 - GPR2 – As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados
 - Este resultado espera que a organização estime o tamanho e/ou complexidade das tarefas e dos artefatos gerados no projeto utilizando métodos adequados, por exemplo, baseado na EAP (Estrutura Analítica do Projeto), em técnicas de estimativa ou dados históricos.
 - GPR3 - O modelo e as fases do ciclo de vida do projeto são definidos
 - Este resultado espera que a organização determine o ciclo de vida para o projeto, indicando suas fases, as relações de sequência, a interdependência entre as fases e os marcos e pontos de controle do projeto.

- GPR4 - (Até o nível F) O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas
 - Este resultado espera que a organização estime o esforço e o custo das tarefas e produtos de trabalho do projeto com base em dados históricos ou métodos de estimativas. As estimativas de esforço e custo são, normalmente, baseadas nos resultados de análises utilizando modelos e/ou dados históricos aplicados ao tamanho, atividades e outros parâmetros de planejamento.
- GPR5 - O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos
 - Este resultado espera que a organização determine o orçamento e o cronograma para projeto, com dependência entre tarefas (incluindo o caminho crítico), e que eles sejam revistos e atualizados, conforme a necessidade ao longo do desenvolvimento.
- GPR6 - Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados
 - Este resultado espera que a organização identifique uma lista de riscos para o projeto e analise esses riscos para determinar o impacto, o grau de importância, a probabilidade e a prioridade de cada risco.
- GPR7 - Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo
 - Este resultado espera que a organização selecione os recursos humanos a partir das competências necessárias para realizar as atividades do projeto, e ainda, quando necessário, sejam planejados treinamentos ou incorporado consultoria externa.
- GPR8 - (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados
 - Este resultado espera que organização planeje a infra-estrutura necessária para executar cada tarefa do projeto, e à disponibilize ao longo do projeto.

- GPR9 - Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança
 - Este resultado espera que a organização defina um plano para a gerência de dados listando todos os documentos gerados no projeto, sua distribuição e mídia para armazenamento, proteção (segurança e sigilo) e recuperação desses dados.
- GPR10 - Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos
 - Este resultado espera que todos os planos que afetam o projeto sejam integrados, de forma que a dependência entre eles seja identificadas, e que todas as informações do planejamento possam ser acessadas a partir do plano de projeto.
- GPR11 - A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados
 - Este resultado espera que a viabilidade do projeto seja avaliada, a partir da visão geral dos objetivos e características dos resultados pretendidos, dos recursos financeiros, técnicos, humanos, bem como restrições impostas pelo cliente, ambiente externo e interno e condições de desenvolvimento.
- GPR12 - O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido
 - Esse resultado espera que todos os interessados tomem conhecimento, revisem e se comprometam com o planejamento do projeto.
- GPR13 - O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado
 - Este resultado espera que o projeto seja monitorado e controlado, comparando o planejado e o realizado, em relação aos aspectos relacionados às tarefas, estimativas, orçamento, e cronograma.

- GPR14 - Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado
 - Este resultado espera que o projeto seja monitorado e controlado, comparando o planejado e o realizado, em relação aos aspectos referentes a recursos materiais e recursos humanos.
- GPR15 - Os riscos são monitorados em relação ao planejado
 - Este resultado espera que os riscos do projeto sejam monitorados, verificando se os parâmetros dos riscos identificados foram alterados, e que alterações no planejamento dos riscos sejam comunicadas aos interessados quando pertinente.
- GPR16 - O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido
 - Este resultado espera que os interessados relevantes do projeto sejam identificados, as fases em que eles são relevantes e como eles serão envolvidos sejam determinados, e que o envolvimento planejado seja monitorado e mantido durante todo o projeto.
- GPR17 - Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento
 - Este resultado espera que revisões nos marcos do projeto sejam planejadas e executadas para verificar, de forma ampla, o andamento de todo o projeto, independente do acompanhamento do dia-a-dia.
- GPR18 - Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas
 - Este resultado espera que os problemas e desvios em relação ao planejamento identificados em atividades de revisões e monitoramento sejam analisados e registrados.

- GPR19 - Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão
 - Este resultado espera que ações corretivas para corrigir os problemas identificados em atividades de revisões e monitoramento, e que foram analisados e registrados, sejam estabelecidas e acompanhadas até serem concluídas.

Gerência de Requisitos

- Propósito:
 - Gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.
- Resultados esperados:
 - GRE1 - O entendimento dos requisitos é obtido junto aos fornecedores de requisitos
 - Este resultado espera que as pessoas autorizadas a definir e alterar os requisitos sejam identificadas, que os requisitos do projeto sejam documentados de alguma forma, e que os requisitos sejam avaliados e aceitos pelas pessoas autorizadas.
 - GRE2 - Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido
 - Este resultado espera que sejam definidos critérios para a avaliação dos requisitos, que os requisitos sejam avaliados por meio dos critérios previamente definidos, e que o comprometimento de todos os membros da equipe técnica com os requisitos seja obtido.
 - GRE3 - A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida
 - Este resultado espera que seja criado e mantido, ao longo do projeto, um mecanismo que permita que os requisitos sejam rastreados horizontal e verticalmente, ou seja, é possível partir de um requisito e chegar ao

documento que descreve a necessidade que o originou e também partindo deste mesmo requisito é possível chegar ao código criado para atendê-lo.

- GRE4 - Revisões em planos e produtos de trabalho do projeto são realizadas visando a identificar e corrigir inconsistências em relação aos requisitos
 - Este resultado espera que sejam realizadas revisões ou algum mecanismo equivalente para identificar inconsistências entre os requisitos e os demais produtos de trabalho do projeto, que as inconsistências identificadas sejam registradas e que ações corretivas sejam executadas a fim de resolvê-las.
- GRE5 - Mudanças nos requisitos são gerenciadas ao longo do projeto
 - Este resultado espera que a solicitações de mudança nos requisitos sejam registradas e que o impacto da mudança seja analisado e aprovado junto aos fornecedores de requisitos antes da sua implementação.

Os demais processos e seus propósitos para todos os níveis de maturidade do MPS.BR podem ser conferidos no Guia Geral (SOFTEX, 2011a).

2.2 JOGOS

Nesta seção são apresentadas, a definição, características e classificações de jogos, a utilização de jogos com fins pedagógicos, como os jogos podem ser projetos com fins pedagógicos, e são apresentadas duas abordagens para avaliação de jogos educacionais.

2.2.1 Definição

Um jogo é uma atividade física ou mental que envolve competição, seja contra outra(s) pessoa(s), contra a máquina ou contra a si mesmo, exigindo que o(s) participante(s) siga(m) um conjunto de regras específicas para atingir um objetivo (HOGLE, 1996). Gibson, Aldrich e Prensky (2007), definem jogos como uma atividade competitiva que é criativa e agradável em sua essência, que é delimitada por determinadas regras e requer certas habilidades (GIBSON, ALDRICH & PRENSKY, 2007).

2.2.2 Características

Segundo Prensky (2001) os jogos devem possuir seis características fundamentais para que eles possam realmente ser caracterizados como jogos, que são:

- **Regras:** é o que diferenciam os jogos dos outros tipos de brinquedos, pois elas impõem limites, elas forçam os jogadores a pegar caminhos específicos para alcançar os objetivos e garantem que todos os jogadores peguem os mesmos caminhos;
- **Metas ou Objetivos:** é o que faz o jogador a jogar;
- **Resultados e Feedback:** é como o progresso em relação ao atendimento dos objetivos é medido;
- **Conflito/Competição/Desafio/Oposição:** são os problemas do jogo que o jogador deve resolver;
- **Interação:** um jogo é dito interativo se ele fornece algum tipo de *feedback* para as ações do jogador; e
- **Representação ou história:** é o que causa o elemento fantasia no jogo.

2.2.3 Classificações

Os jogos digitais podem ser classificados em oito gêneros diferentes, que geralmente se sobrepõem, que são (PRENSKY, 2001; HERZ, 1997):

- **Ação:** são jogos com ênfase no movimento. Esta categoria inclui os jogos conhecidos como *side scroller*, são jogos em que os personagens se movem do lado esquerdo para o lado direto da tela, labirinto, jogos de pular plataforma, jogos em que caem coisas que o jogador tem que atirar, corridas de carro, e perseguição. Esta categoria também inclui os jogos conhecidos como *shoot 'em up*, que são jogos onde o jogador controla um personagem solitário, geralmente em uma espaçonave ou aeronave, disparando contra um grande número de inimigos, e desviando de seus ataques;
- **Aventura:** são jogos onde o jogador deve encontrar um caminho em um mundo desconhecido, pegar objetos, e resolver desafios;

- **Luta:** são jogos onde dois personagens lutam até que um deles seja derrotado;
- **Puzzle:** são jogos onde há problemas a serem resolvidos, tipicamente visuais e sem qualquer história;
- **RPG – Role Playing Game:** são jogos em que os jogadores assumem os papéis de personagens e criam narrativas colaborativamente. O progresso de um jogo se dá de acordo com um sistema de regras predeterminado, dentro das quais os jogadores podem improvisar livremente. As escolhas dos jogadores determinam a direção que o jogo irá tomar;
- **Simulação:** são jogos onde o jogador normalmente pilota, dirige, constrói ou mantém alguma coisa;
- **Esporte:** são jogos onde o conteúdo ao invés da jogabilidade é um fator determinante, e geralmente são jogos de ação e simulação que permitem controlar um ou mais jogadores ao mesmo tempo; e
- **Estratégia:** são jogos onde o jogador normalmente é responsável por alguma coisa (um exército ou uma civilização), evoluindo-a da maneira que desejar para ele mesmo ou para ganhar dos oponentes.

2.2.4 Jogos Educacionais

Muitos autores defendem a utilização de jogos em sala de aula, principalmente por facilitar a aprendizagem (PASSERINO, 1998; PRENSKY, 2001; GIBSON, ALDRICH & PRENSKY, 2007). Segundo Passerino (1998), essa facilidade vem de algumas características dos jogos tais como: a capacidade de absorver o jogador de maneira intensa e total, com um clima de entusiasmo, sentimento de exaltação e tensão seguidas por um estado de alegria e distensão; o envolvimento emocional durante a progressão do jogo; a espontaneidade e a criatividade liberada pelos jogadores; a limitação de tempo e espaço, onde o jogo tem um início um meio e um fim e um caráter dinâmico, sendo reservado, independente da forma que assuma, como um mundo temporário e fantástico; a existência de regras rigorosamente definidas. Cada jogo evolui de acordo com regras que determinam o que pode ou não dentro do mundo imaginário daquele jogo, auxiliando no processo social dos discentes. A estimulação da imaginação, da auto-afirmação e da autonomia que os jogos proporcionam (PASSERINO, 1998; FIGUEIREDO & FIGUEIREDO, 2011).

A utilização de jogos no ensino tem sido originada principalmente pelo alto fator motivacional que esta ferramenta pode trazer para aprendizagem (GIBSON, ALDRICH & PRENSKY, 2007). As quatro características dos jogos que contribuem para aumentar a motivação da aprendizagem são: desafio, fantasia, curiosidade e controle. O desafio mantém os estudantes comprometidos com a atividade, a fantasia aumenta o entusiasmo dos estudantes providenciando um cenário imaginário e atraente para os mesmos, a curiosidade despertada pelo novo cenário estimula os estudantes a explorar o desconhecido, e o controle fornece a sensação de alto-determinação aos aprendizes (GIBSON, ALDRICH & PRENSKY, 2007).

Segundo Prensky (2001), o ensino baseado em jogos funciona primeiramente por três motivos:

1. O primeiro é o comprometimento adicional que vem de colocar a aprendizagem no contexto de um jogo;
2. O segundo é o processo de aprendizagem interativo aplicado, que pode e deve assumir diferentes formas de acordo com os objetivos educacionais; e
3. O terceiro é a maneira como os dois são colocados juntos, onde há diferentes maneiras de fazer isto, e a melhor solução é altamente contextual.

O desenvolvimento de um jogo educacional deve levar em consideração uma série de fatores, que são: a audiência, ou seja, qual é o público alvo do jogo a ser desenvolvido; o assunto que será ensinado; o contexto em que ele estará inserido; a tecnologia disponível; os recursos e a experiência que pode ser realizada; e como o jogo será distribuído (PRENSKY, 2001). A Tabela 4 apresenta atividades de aprendizagem e estilos de jogos de acordo com tipos de conteúdo ensinado.

Tabela 4. Conteúdos x atividade de aprendizagem x estilos de jogos

Conteúdo	Exemplos	Atividades de Aprendizado	Estilos de Jogos Possíveis
Fatos	Leis, políticas, especificações de produto	Perguntas, memorização, associação, exercício	Competições, Mnemônicos, Cartas, Ação, Esporte
Competências	Entrevista, ensino de venda, manuseio de uma máquina, gerenciamento de projetos	Imitação, experiência de treinamento, prática contínua, desafio crescente	Estado persistente, RPG, aventura, detetive
Julgamento	Tomadas de decisão, cronometragem, ética, contratação	Revisão de casos, realização de perguntas, realização de escolhas (prática), experiências, treinamento	RPG, detetive, interação entre vários jogadores, aventura, estratégia
Comportamentos	Supervisão, exercitando o auto-controle, dando exemplos	Imitação, experiência, treinamento, prática	RPG
Teorias	Lógica de mercado, como as pessoas aprendem	Lógica, experimentação, questionamentos	Simulação, jogos de construção, teste de realidade
Raciocínio	Pensamento estratégico e tático, análise de qualidade	Problemas, exemplos	<i>Puzzles</i>
Processo	Auditoria, criação de estratégia	Análise de sistema e desconstrução prática	Estratégia, aventura, simulação
Procedimentos	Montagem, procedimentos legais de caixa de banco	Imitação, prática	Jogos cronometrados, jogos de reflexo
Criatividade	Invenção, design de produto	Jogo, memorização	<i>Puzzle</i> , jogos de invenção
Linguagem	Siglas, línguas estrangeiras, negócios ou jargões profissionais	Imitação, prática contínua, imersão	RPG, jogos de reflexo, jogos de carta
Sistemas	Cuidados da saúde, mercados, refinarias	Compreender os princípios, tarefas graduadas, jogando em micro mundos	Jogos de simulação
Observação	Humor, moral, ineficiências, problemas	Observação, experiência	Jogos de concentração, aventura
Comunicação	Linguagem apropriada, cronometragem, envolvimento	Imitação, prática	RPG, jogos de reflexo

Fonte: Adaptado de Prensky (2001)

2.2.5 Projeto de Jogos Educacionais

Para ajudar os designers de jogos ou profissionais da educação a projetarem jogos educacionais que permitam uma aprendizagem eficaz, um *framework* que combina as teorias de aprendizagem e pedagogia com requisitos de jogos foi elaborado (YUSOFF *et al.*, 2009). A Figura

5 apresenta o *framework* para desenvolvimento de jogos no formato de um diagrama de classes, e seus componentes são discutidos a seguir.

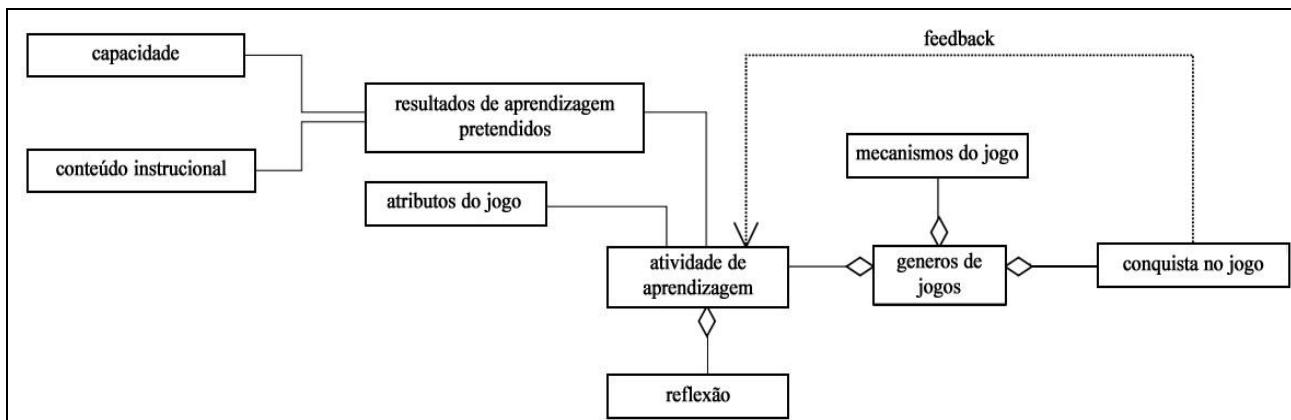


Figura 5. *Framework* para desenvolvimento de jogos educacionais

Fonte: Adaptado de Yusoff e outros (2009)

A capacidade refere-se ao cognitivo, psicomotor e possíveis habilidades efetivas que o aluno deve desenvolver como resultado de jogar o jogo. Habilidades cognitivas incluem as capacidades de recuperação, análise, síntese e avaliação. Habilidades psicomotoras incluem as capacidades de execução de forma fluente e no tempo correto. Habilidades efetivas incluem as capacidades de identificação, adoção e valorização de atitudes adequadas e pontos de vista (YUSOFF *et al.*, 2009).

O conteúdo instrucional refere-se ao assunto que o aluno deve aprender. O real conteúdo ou o tipo de conteúdo que os alunos devem aprender pode ser uma lista exaustiva (YUSOFF *et al.*, 2009). Gilbert e Gale (Gilbert & Gale, 2008 *apud* YUSOFF *et al.*, 2009) afirmam que os conteúdos podem ser classificados em quatro tipos: fatos, procedimentos, conceitos e princípios.

Os resultados de aprendizagem são os objetivos a serem alcançados como resultado de jogar o jogo. Um resultado de aprendizagem pretendido é uma combinação particular de assunto e capacidade (YUSOFF *et al.*, 2009), por exemplo, um aluno deve ser capaz de recordar a data do descobrimento do Brasil, ou deve ser capaz de analisar se um animal em particular é um réptil.

Os atributos do jogo são os aspectos de um jogo que contribuem para o aprendizado e motivação do aluno. Esses atributos são apresentados na Seção 2.2.6.2.

A atividade de aprendizagem é a atividade destinada a manter o aluno motivado e aprendendo no mundo do jogo. O profundo envolvimento ou imersão por parte do aluno depende da

eficácia na prestação dessas atividades. Por exemplo, se um aluno precisa ser capaz de recordar um conceito, a atividade de aprendizagem deve mostrar um exemplo do conceito e perguntar para o aluno o nome do conceito, seguido por um *feedback* na resposta (YUSOFF *et al.*, 2009).

A reflexão é onde o aluno pensa sobre o propósito das atividades de aprendizagem que têm sido desenvolvidas, e decide a estratégia a ser aplicada durante a próxima atividade. Deve haver um lugar dentro do jogo para a reflexão sem deixar que o aluno saia do mundo do jogo, isso pode ser feito através da oferta de atividades de reflexão dentro do jogo (YUSOFF *et al.*, 2009).

O gênero é o tipo ou categoria de um jogo. Segundo Prensky (2001) os jogos digitais podem ser classificados em oito gêneros diferentes, que são: ação, aventura, luta, *puzzle*, RPG, simulação, esporte e estratégia.

A mecânica do jogo e as suas regras definem os detalhes do jogo. Por exemplo, se o gênero do jogo for estratégia em tempo real, então ele pode exigir a mecânica de gestão de recursos e controle de território. As atividades de aprendizagem e o conteúdo instrucional influenciam no mecanismo que será utilizado no jogo, a fim de projetar um jogo que irá melhor atender um determinado estilo de aprendizagem, o aluno alvo específico, ou um conjunto particular de resultados pretendidos (YUSOFF *et al.*, 2009).

E por fim a conquista no jogo representa o desempenho do aluno no jogo. Essa conquista pode ser indicada pelos escores dos jogos, a quantidade total de recursos ou bens recolhidos no âmbito do jogo, ou o tempo necessário para atingir os objetivos do jogo. Além disso, ele fornece o prazer da recompensa para o aluno, e pode ter o propósito de avaliação dos alunos (YUSOFF *et al.*, 2009).

2.2.6 Avaliação de Jogos Educacionais

Esta seção apresenta duas abordagens para avaliação de jogos educacionais, uma baseada em dados quantitativos e outra baseada em dados qualitativos. Essas abordagens serviram de base para o planejamento e execução da avaliação do jogo desenvolvido neste trabalho.

2.2.6.1 Framework para Avaliação Empírica de Jogos Educacionais

O *framework* para avaliação empírica de jogos educacionais (KOCHANSKI, 2009) foi desenvolvido para facilitar o planejamento, aplicação e replicação de experimentos para avaliação dos efeitos de aprendizagem do uso de jogos educacionais na área de Engenharia de Software. O *framework* é dividido em cinco partes: Parte I – Definição do Experimento; Parte II – Planejamento do Experimento; Parte III – Operação do Experimento; Parte IV – Análise e Interpretação dos Dados; e Parte V – Apresentação dos resultados (KOCHANSKI, 2009).

A primeira parte do *framework* trata das definições iniciais de como o experimento será realizado, e é subdividida em (KOCHANSKI, 2009):

1. Estratégia de Pesquisa:
 - a) Quantitativa: quando a forma de medição utilizada é baseada em valores numéricos precisos, visa obter como resultado quantidades e é orientada a verificação.
 - b) Qualitativa: quando a forma de medição é baseada na forma naturalista, geralmente buscando como resultados qualificadores e é orientada a descoberta.
2. Forma de Realização:
 - a) *In-vitro*: quando as condições para realização do estudo tiverem que ser criadas.
 - b) *In-vivo*: quando o estudo pode se realizado no ambiente e forma como realmente ocorre.
 - c) *In-virtuo*: quando o estudo envolve a interação dos participantes com modelos da realidade simulados em computador.
 - d) *In-silico*: quando o estudo é totalmente composto por modelos numéricos.
3. Abordagem de Pesquisa:
 - a) Analítica: quando o propósito da pesquisa for a obtenção de um entendimento inicial sobre determinada questão ou fenômeno para posterior refinamento da pesquisa.
 - b) Descritiva: quando os dados forem quantitativos e se pretenda descrever algum fenômeno.

- c) Explanativa: quando o propósito de pesquisa for explanar porque determinado evento ocorre e construir, elaborar, estender ou testar uma teoria.
- 4. Estratégia para a seleção de grupos: quando for necessária a divisão dos participantes em diferentes grupos, é importante que seja definida uma estratégia que garanta a aleatoriedade da seleção dos indivíduos que farão parte de um ou outro grupo.
- 5. Questionários: descreve os questionários que serão aplicados durante o experimento, exemplos: concordância e perfil.
- 6. Pré-condições: descreve os conhecimentos prévios que as pessoas devem possuir para que elas tenham condições de participar do experimento de forma adequada.
- 7. Design Instrucional: descreve o plano de instrução que será realizado com os participantes do experimento.

Na segunda parte do *framework* o objetivo principal é garantir o mínimo de desvios na execução do experimento, e é subdividida em (KOCHANSKI, 2009):

- 1. Seleção do Contexto: determina o ambiente em que o estudo será realizado.
- 2. Definição das Hipóteses: deve ser descrita a hipótese nula (H_0) e a negação da hipótese nula (H_1), bem como as demais hipóteses de pesquisa existentes.
- 3. Variáveis de Controle: determina as variáveis que serão medidas e que proverão os principais dados para análise e interpretação de resultados.
- 4. Seleção dos Participantes: descreve o perfil das pessoas que poderão participar do experimento.
- 5. Design de Experimento Utilizado:
 - a) Apenas pós-teste: quando existem dois grupos ou o grupo pode ser dividido em dois, a seleção dos participantes de cada grupo é aleatória, a medição será realizada apenas no final, existem medições para cada grupo, e existe uma avaliação do efeito do tratamento.
 - b) Pré-teste e pós-teste: quando existem dois grupos ou o grupo pode ser dividido em dois, a seleção dos participantes de cada grupo é aleatória, será realizada uma medição no início, é realizada uma intervenção controlada e é realizada uma

medição no final, existem duas medições em cada grupo, uma antes do tratamento e outra depois do tratamento, e existe uma avaliação do efeito do tratamento.

6. Planejamento da Instrumentação: descreve qual será o tratamento ou intervenção realizada e como tal tratamento ou intervenção ocorrerá, quais serão os objetos ou equipamentos utilizados, as diretrizes de realização do experimento e os instrumentos de medição.
7. Avaliação da Viabilidade: devem ser relacionadas as ameaças identificadas, conforme o tipo de ameaça à validade, bem como uma descrição detalhada da ameaça para que a mesma possa ser discutida na fase de análise e interpretação.

Na terceira parte do *framework* devem ser registradas informações de como os alunos serão informados sobre o experimento, quais serão os materiais utilizados, como o comprometimento será garantido, como os dados serão coletados, como o ambiente será organizado, como será garantido que os participantes entenderão o seu papel, e como será garantida a conformidade com os procedimentos planejados para o experimento (KOCHANSKI, 2009).

Na quarta parte do *framework* são apresentadas as informações para a posterior análise e interpretação dos dados, e é subdividida em (KOCHANSKI, 2009):

1. Parâmetros Populacionais:
 - a) Teste estatístico paramétrico: quando os dados forem extraídos de populações com distribuição de probabilidade especificada, a distribuição dos valores da variável de pesquisa for normal, a escala da variável de pesquisa for intervalar ou ratio, e a variável de pesquisa permite a realização de operações aritméticas a partir dos valores das amostras.
 - b) Teste estatístico não-paramétrico: quando não é necessário conhecer a distribuição da variável de pesquisa na população, o tamanho da população amostrada for pequena, e a escala da variável de pesquisa for ordinal ou nominal.
2. Teste Utilizado:
 - a) Teste t independente: quando o tipo de teste for paramétrico, a comparação dos dados for realizada de forma agrupada, e o objetivo for comparar relações entre dois grupos.

- b) Teste t relacionado: quando o tipo de teste estatístico for paramétrico, os dois conjuntos de valores estiverem correlacionados de forma significativa, e existe um alto coeficiente de correlação entre dois conjuntos de dados.
 - c) Mann-Whitney: quando o tipo de teste estatístico for não-paramétrico, as amostras são independentes, as amostras foram obtidas a partir da mesma população, a observação das amostras é independente, e as observações são comparáveis.
 - d) Wilcoxon: quando o tipo de teste estatístico for paramétrico, as amostras são relacionadas, as amostras foram obtidas a partir da mesma população, a observação das amostras é independente, e as observações são comparáveis.
3. Estatística Utilizada: descrição detalhada de como os dados serão calculados e analisados.
 4. Redução do Conjunto de Dados: descrição da estratégia a ser utilizada para a análise e redução do conjunto de dados.
 5. Teste de Hipóteses: descrição da estratégia a ser utilizada para testar as hipóteses de pesquisa.

Na quinta parte do *framework* deve ser realizada a apresentação dos resultados do experimento. Para padronizar a apresentação de tais resultados é proposta a seguinte estrutura de apresentação de resultados (KOCHANSKI, 2009):

1. Resumo: deve apresentar o objetivo geral da pesquisa, o método utilizado, os resultados e as conclusões do trabalho, formando uma seqüência corrente de frases concisas.
2. Introdução: deve apresentar a definição clara, concisa e objetiva do tema, e da delimitação precisa das fronteiras do estudo em relação ao problema e aos objetivos a serem estudados.
3. Declaração do Problema: devem ser apontados os propósitos e as linhas gerais que orientaram a pesquisa, ou seja, apresentar o problema ou tema central da pesquisa, contextualizando, destacando sua importância e seus limites quanto à extensão e à profundidade.

4. Desempenho do Experimento: devem ser apresentados os conceitos relacionados com a pesquisa, descrição dos conceitos considerados para a realização do experimento, como o experimento foi conduzido e os resultados obtidos através do experimento.
5. Discussão e Conclusões: consiste na revisão sistemática dos resultados e sua discussão, destacando os resultados mais significativos em termos de contribuição para futuros estudos. Esta seção deve apresentar deduções lógicas correspondentes aos propósitos previamente estabelecidos na introdução da apresentação de resultados.
6. Recomendação e Trabalhos Futuros: devem ser realizados os registros de sugestões para extensões do estudo realizado, bem como sugestões de outros estudos que possam ser realizados com base nas idéias geradas a partir do estudo corrente.
7. Agradecimentos: espaço para que sejam citadas as pessoas, grupos ou entidades que contribuíram para a realização do estudo.
8. Referências: deve ser apresentada a relação de referências bibliográficas utilizadas no relatório de apresentação de resultados.
9. Apêndices: espaço destinado para a inclusão de materiais relacionados à apresentação dos resultados e que não fizeram parte do corpo do texto.

2.2.6.2 Modelo de Avaliação de Jogos Educacionais

O Modelo de Avaliação de Jogos Educacionais (SAVI, 2011) foi desenvolvido visando orientar desenvolvedores, pesquisadores e professores a avaliarem a qualidade de jogos educacionais voltados para o ensino e aprendizagem da Engenharia de Software.

O modelo de avaliação é focado na avaliação da reação do aluno ao jogo utilizado, ou seja, a percepção deles em relação a experiência de aprendizagem (SAVI *et al.*, 2010). Segundo os mesmos autores, o modelo procura avaliar se um jogo: (i) consegue motivar os estudantes a utilizarem o recurso como material de aprendizagem; (ii) proporciona uma boa experiência nos usuários (p.ex. se ele é divertido); e (iii) se gera uma percepção de utilidade educacional entre seus usuários (ou seja, se os alunos acham que estão aprendendo com o jogo).

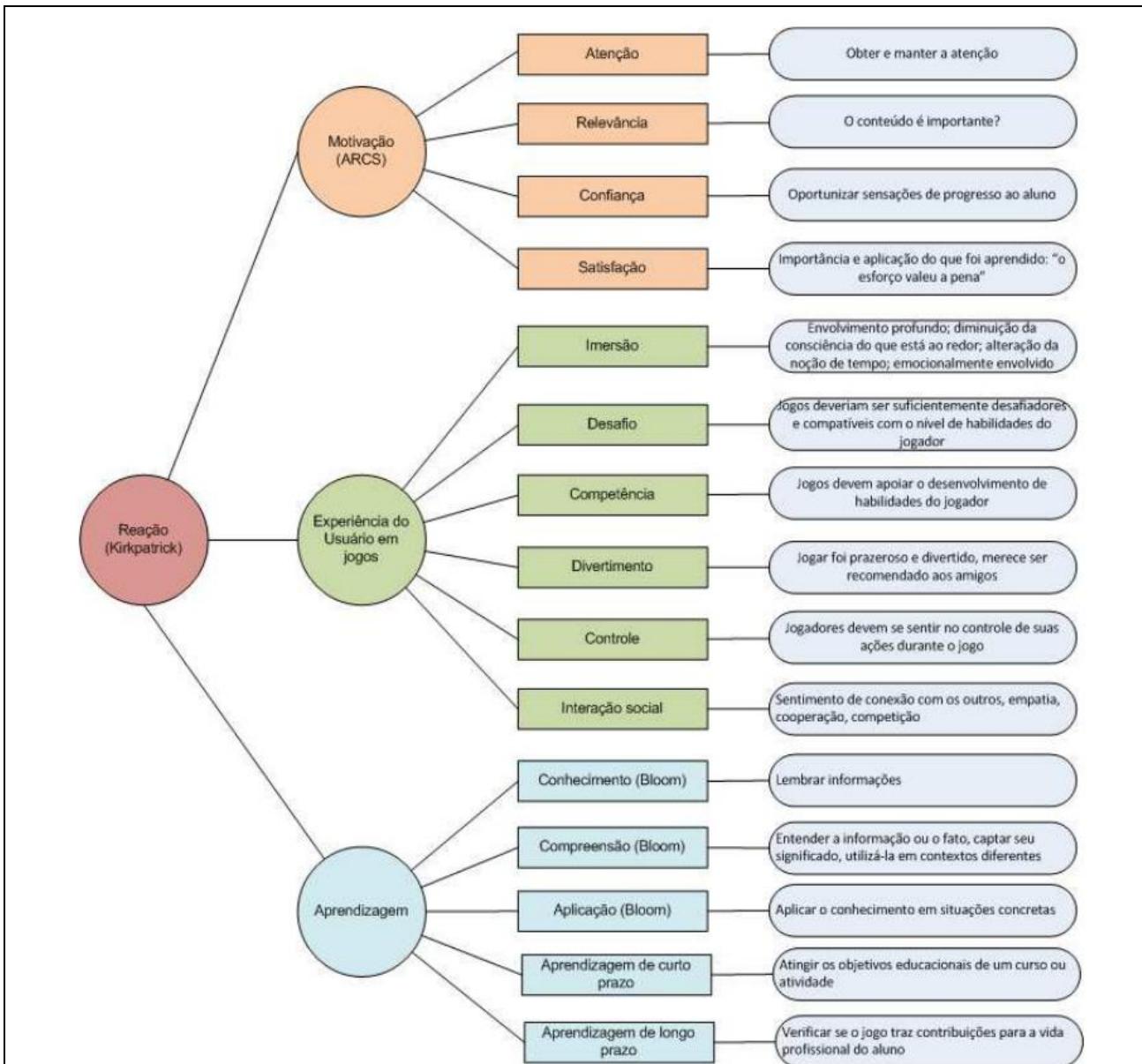


Figura 6. Estrutura do modelo de avaliação de jogos educacionais

Fonte: Savi (2011).

A Figura 6 apresenta a estrutura do modelo de avaliação. Como pode ser observado, o modelo dividi a avaliação da reação do aluno no jogo em três aspectos: motivação, experiência do usuário e aprendizagem. Cada aspecto possui algumas dimensões que devem ser avaliadas. Em relação ao aspecto motivação devem ser avaliadas as seguintes dimensões (SAVI, 2011):

1. Atenção: além de ser um elemento de motivação, é, também, um pré-requisito para a aprendizagem. O desafio é obter e manter um nível satisfatório da atenção dos alunos ao longo de um período de aprendizagem (KELLER, 1987; SAVI, 2011).

2. Relevância: representa o nível de associação que os alunos conseguem perceber entre seus conhecimentos prévios e as novas informações (KELLER, 1987; SAVI, 2011).
3. Confiança: está relacionada em criar expectativas positivas aos estudantes, permitindo que tenham sucesso. Isso pode ser alcançado ao se proporcionar experiências de sucesso no uso do material escolar decorrentes da própria habilidade e esforço dos alunos (p.ex., com exercícios em nível crescente de dificuldade) (KELLER, 1987; SAVI, 2011).
4. Satisfação: está relacionada com os sentimentos positivos dos alunos sobre a experiência de aprendizagem, e isso pode vir com recompensas e reconhecimento. Os estudantes devem sentir que o esforço dedicado aos estudos foi apropriado e que houve consistência entre objetivos, conteúdo e testes de avaliações (KELLER, 1987; SAVI, 2011).

Em relação ao aspecto experiência do usuário devem ser avaliadas as seguintes dimensões (SAVI, 2011):

1. Imersão: está relacionado com a experiência de profundo envolvimento no jogo, que geralmente provoca um desvio de foco do mundo real para o mundo do jogo. A imersão provoca uma distorção da noção de tempo, sendo que muitos jogadores relatam que passam longos períodos jogando sem perceberem o tempo passar (SWEETSER & WYETH, 2005; SAVI, 2011).
2. Desafio: é um dos aspectos mais importantes nos bons jogos. Um jogo precisa ser suficientemente desafiador, estar adequado ao nível de habilidade do jogador, apresentar variações no nível de dificuldade, e manter um ritmo adequado (SWEETSER & WYETH, 2005; SAVI, 2011).
3. Competência: está relacionada com a percepção de habilidades, controle e uso dessas habilidades para explorar o jogo e progredir. Devem haver informações suficientes para que se comece a jogar sem a necessidade da leitura de longos manuais e sem a demanda por longas explicações sobre regras, cenário ou história do jogo (SWEETSER & WYETH, 2005; SAVI, 2011).
4. Divertimento: está relacionado com os sentimentos de diversão, prazer, relaxamento, distração e satisfação que o jogo deve proporcionar, permitindo que jogadores desejem voltar a participar do jogo e indiquem para seus colegas (SILVA, 2011).

5. Controle: está relacionado com a sensação de independência, domínio, autonomia, poder e liberdade. Os jogadores devem poder exercer um senso de controle sobre suas ações, conseguir traduzir suas intenções em comportamentos dentro do jogo, sentir controle sobre os movimentos de seus personagens e a maneira como ele explora o ambiente (SILVA, 2011).
6. Interação Social: está relacionada com o sentimento de compartilhar um ambiente com outras pessoas e de se ter um papel ativo nele. Para possibilitar a interação social os jogos devem criar oportunidades para os jogadores competirem, cooperarem e se conectarem (SWEETSER & WYETH, 2005; SAVI, 2011)

E em relação ao aspecto aprendizagem devem ser avaliadas as seguintes dimensões (SAVI, 2011):

1. Conhecimento: habilidade de lembrar informações e conteúdos previamente abordados como fatos, datas, palavras, teorias, métodos, classificações, lugares, regras, critérios, procedimentos, etc (ANDERSON & KRATHWOHL, 2001; SAVI, 2011).
2. Compreensão: habilidade de compreender e dar significado ao conteúdo. Essa habilidade pode ser demonstrada por meio da tradução do conteúdo compreendido para uma nova forma (oral, escrita, diagramas, etc.) ou contexto (ANDERSON & KRATHWOHL, 2001; SAVI, 2011).
3. Aplicação: habilidade de usar informações, métodos e conteúdos aprendidos em novas situações concretas. Isso pode incluir aplicações de regras, métodos, modelos, conceitos, princípios, leis e teorias (ANDERSON & KRATHWOHL, 2001; SAVI, 2011).
4. Aprendizagem de Curto Término: baseada nos objetivos educacionais mais imediatos de um curso, atividade ou material (SAVI, 2011).
5. Aprendizagem de Longo Término: busca verificar se o curso ou atividade trazem contribuição para a vida profissional (SAVI, 2011).

Para avaliar esses aspectos o modelo possui um questionário para coletar dados sobre cada uma dessas dimensões. O questionário é composto por afirmações na qual os participantes devem indicar o quanto eles concordam com elas, de acordo com a escala de Likert, variando de “discordo fortemente” (-2) até “concordo totalmente” (+2) (SAVI, 2011). Esse questionário é apresentado no

Apêndice G e a Tabela 6 apresenta as questões associadas com cada dimensão do modelo de avaliação.

Tabela 5. Questões por dimensão do modelo de avaliação de jogos educacionais

Dimensões	Questões
Atenção	Q1, Q2 e Q3
Relevância	Q4, Q5 e Q6
Confiança	Q7 e Q8
Satisfação	Q9 e Q10
Imersão	Q11, Q12 e Q13
Interação social	Q14, Q15 e Q16
Desafio	Q17 e Q18
Competência	Q19, Q20, Q21 e Q22
Divertimento	Q23 e Q24
Conhecimento	Q28
Compreensão	Q29
Aplicação	Q30
Aprendizagem de curto termo	Q25 e Q26
Aprendizagem de longo termo	Q27

Para apoiar a etapa de análise e interpretação dos resultados o modelo possui uma planilha elaborada para ser utilizada em avaliações utilizando este modelo. Esta planilha auxilia na organização das informações e geração automática dos gráficos com os resultados, e está disponível para download em (SAVI, 2012).

3 MAPEAMENTO SISTEMATICO

Mapeamento sistemático é um tipo específico de revisão sistemática, que utiliza a mesma metodologia de base da revisão sistemática, porém é focado em questões mais amplas, de natureza exploratória, cujo propósito é dar uma visão geral de uma área de pesquisa (KITCHENHAM, 2007). Este mapeamento sistemático buscou evidências para avaliar como o conteúdo dos principais modelos de capacidade de software tem sido explorado pelos jogos educacionais.

3.1 PLANEJAMENTO

Esta seção apresenta o planejamento do mapeamento sistemático realizado.

3.1.1 Perguntas de Pesquisa

As perguntas de pesquisa são a parte mais importante do mapeamento, pois é o que deve ser respondido ao final do mesmo (KITCHENHAM, 2007). A pergunta de pesquisa que orientou este mapeamento é apresentada na Tabela 6.

Tabela 6. Pergunta de pesquisa do mapeamento sistemático

Pergunta 1: Como o conteúdo dos principais modelos de capacidade de software tem sido explorado pelos jogos educacionais existentes?	
População	Jogos educacionais (computacionais ou não) com foco no ensino de Engenharia de Software e/ou Melhoria de Processo de Software e os modelos de capacidade de software CMMI (SEI, 2010), MPS.BR (SOFTEX, 2011a) e ISO/IEC 15504 (2003).
Intervenção	Conteúdo explorado pelos jogos ou ferramentas de simulação.
Comparação	Conteúdo dos modelos CMMI, MPS.BR, ISO/IEC 15504.
Resultados	Grau de cobertura, densidade (distribuição das áreas exploradas pelos jogos) e número de jogos existentes para cada área.
Contexto	Capacitação em Melhoria de Processo de Software.

3.1.2 Estratégia de Busca

Esta seção apresenta a estratégia de busca utilizada na procura dos estudos primários relacionados com a pergunta de pesquisa.

3.1.2.1 Palavras Chave

Foram utilizadas nesta pesquisa palavras nos idiomas português e inglês. O idioma inglês foi utilizado por ser o idioma padrão das fontes de pesquisa conveniadas ao portal da CAPES, e o idioma português por ser o idioma padrão do país em que esta pesquisa foi realizada. O idioma português foi utilizado somente em pesquisas no Google Scholar. A Tabela 7 apresenta as palavras chaves utilizadas nesta pesquisa.

Tabela 7. Palavras chave utilizadas no mapeamento sistemático

Palavras Chave	
Em inglês	CMMI, education, game, ISO/IEC 15504, learning, MPS.BR, software configuration management, software construction software design, Software Engineering, software maintenance, software measurement, software process, software process improvement, software project, software quality, software requirements, software testing, teaching, e training.
Em português	Aprendizagem, CMMI, construção de software, design de software, educação, Engenharia de Software, ensino, gerência de configuração de software, ISO/IEC 15504, jogo, manutenção de software, medição de software, melhoria de processo de software, MPS.BR, processo de software, projeto de software, qualidade de software, requisitos de software, teste de software, e treinamento.

3.1.2.2 Fontes de Pesquisa

As fontes de pesquisa utilizadas foram as principais bibliotecas digitais conveniadas ao portal da CAPES e o Google Scholar. As fontes de pesquisa utilizadas estão listas a seguir:

- IEEEExplore <<http://ieeexplore.ieee.org>>
- ACM Digital Library <<http://portal.acm.org>>
- Citeseer library <<http://citeseerx.ist.psu.edu>>
- SpringerLink <<http://www.springerlink.com>>
- WILEY Interscience <<http://onlinelibrary.wiley.com>>
- ScienceDirect <<http://www.sciencedirect.com>>
- Google Scholar <<http://scholar.google.com.br>>

3.1.2.3 Termos de Busca

Esta seção apresenta os termos de busca utilizados em cada fonte de pesquisa. Os termos de busca foram elaborados utilizando as palavras chaves apresentadas na Seção 2.2.2 e obedecendo a sintaxe de cada fonte de pesquisa. Os termos de busca utilizados nesta pesquisa são apresentados na Tabela 8.

Tabela 8. Termos de busca do mapeamento sistemático

Fonte	Termos
IEEEExplore	(game) AND (education OR learning OR teaching OR training OR teach OR learn) AND ("software engineering" OR "software project" OR "software requirements" OR "software design" OR "software construction" OR "software testing" OR "software maintenance" OR "software configuration management" OR "requirements engineering" OR "software process" OR "software measurement" OR "software quality" OR "software process improvement" OR "CMMI" OR "MPS.BR" OR "15504" OR "12207") Publication Year: From 1990 to 2012.
ACM Digital Library	(Abstract:((game) and (education or learning or teaching or training or teach or learn) and ("software engineering" or "software project" or "software requirements" or "software design" or "software construction" or "software testing" or "software maintenance" or "requirements engineering" or "software configuration management" or "software process" or "software measurement" or "software quality" or "software process improvement" or "CMMI" or "MPS.BR" or "15504" or "12207"))) Published since 1990
Citeseer Library	abstract:(game (education OR learning OR teaching OR training OR teach OR learn) ("software engineering" OR "software project" OR "software requirements" OR "software design" OR "software construction" OR "software testing" OR "software maintenance" OR "software configuration management" OR "requirements engineering" OR "software process" OR "software measurement" OR "software quality" OR "software process improvement" OR "CMMI" OR "MPS.BR" OR "15504" OR "12207")) AND year:[1990 TO 2012]
Springer Link	ab:(game and ("software engineering" or "software project" or "software requirements" or "software design" or "software construction" or "software testing" or "software maintenance" or "software configuration management" or "requirements engineering")) ab:(game and ("software process" or "software measurement" or "software quality" or "software process improvement" or "CMMI" or "MPS.BR" or "15504" or "12207"))
WILEY Interscience	game in Abstract AND education OR learning OR teaching OR training OR teach OR learn in Abstract AND "software engineering" OR "software project" OR "software requirements" OR "software design" OR "software construction" OR "software testing" OR "software maintenance" OR

	"software configuration management" OR "requirements engineering" OR "software process" OR "software measurement" OR "software quality" OR "software process improvement" OR "CMMI" OR "MPS.BR" OR "15504" OR "12207" in Abstract between years 1990 and 2012
ScienceDirect	pub-date > 1989 and TITLE-ABSTR-KEY(game AND(education OR learning OR teaching OR training OR teach OR learn)AND ("software engineering" OR "software project" OR "software requirements" OR "software design" OR "software construction" OR "software testing" OR "software maintenance" OR "software configuration management" OR "requirements engineering" OR "software process" OR "software measurement" OR "software quality" OR "software process improvement" OR "cmmi" OR "MPS.BR" OR "15504" OR "12207"))
Google Scholar	game (education OR learn OR teach OR training) ("software engineering" OR "software project" OR "software requirements" OR "software design" OR "software construction" OR "software testing" OR "software maintenance" OR "requirements engineering") game (education OR learn OR teach OR training) ("software process" OR "software quality" "software acquisition" OR "CMMI" OR "MPS.BR" OR "15504" OR "12207") jogo (educação OR aprendizagem OR ensino OR treinamento) ("Engenharia de Software" OR "Projeto de Software" OR "Requisitos de Software" OR "Projeto de Software" OR "Construção de Software" OR "teste de software" OR "engenharia de requisitos") jogo (educação OR aprendizagem OR ensino OR treinamento) ("processo de software" OR "qualidade de software" OR "aquisição de software" OR "CMMI" OR "MPS.BR" OR "12207" OR "15504") Filtros: - Publicados entre 1990-2012. - Com pelo menos um resumo.

3.1.3 Seleção dos Trabalhos

Nesta seção são apresentados os critérios de inclusão e exclusão, assim como o procedimento utilizado na seleção dos trabalhos relevantes.

3.1.3.1 Critérios de Inclusão e Exclusão dos Trabalhos

Foram examinados todos os artigos sobre jogos para o ensino de Engenharia de Software ou melhoria de processo de software disponível na Web (indicados por buscas em base de dados científicas e no Google Schoolar), publicados entre 1990 e 2010. Foram considerados todos os tipos

de jogos com objetivos educacionais, incluindo jogos de computador ou não (por exemplo: jogos de tabuleiro, cartas, etc.). Foram excluídos os resultados que:

- Descreviam apenas exercícios que não envolviam simulação;
- Descreviam apenas protótipos conceituais de jogos, que foram apenas modelados e ainda não chegaram a ser implementados;
- Não tenham sido publicados em revistas especializadas, em anais de congressos, simpósios, seminários ou não eram trabalhos de conclusão de curso ou dissertações;
- Não tratavam de jogos com finalidades acadêmicas ou educacionais;
- Não forneciam informação suficiente sobre o jogo; e
- Não eram voltados para o ensino de Engenharia de Software ou melhoria de processo de software.

3.1.3.2 Procedimento de Seleção

A seleção dos trabalhos foi realizada em três etapas:

- a) **Pesquisa dos trabalhos:** A pesquisa dos trabalhos foi feita aplicando os termos de busca nas fontes de pesquisa selecionadas.
- b) **Seleção dos trabalhos relevantes (1º filtro):** No primeiro filtro da seleção dos trabalhos relevantes foram conferidos os títulos e os resumos dos trabalhos retornados na pesquisa, verificando sua relevância de acordo com os critérios de inclusão e exclusão. Trabalhos irrelevantes ou duplicados foram removidos.
- c) **Seleção dos trabalhos relevantes (2º filtro):** No segundo filtro da seleção dos trabalhos relevantes, os trabalhos que passaram pelo primeiro filtro foram lidos completamente verificando novamente sua relevância de acordo com os critérios de inclusão e exclusão. Trabalhos irrelevantes foram removidos.

3.1.4 Estratégia de Extração dos Dados

Esta seção apresenta as informações extraídas dos trabalhos selecionados necessárias para responder a pergunta de pesquisa apresentadas na Seção 3.1.1 . Em cada artigo selecionado foram extraídas as seguintes informações:

- **Identificação do trabalho:** identificador do trabalho composto por um número seqüencial;
- **Nome:** Nome do jogo relatado no estudo;
- **Referências:** referência do trabalho, bem como documentos adicionais que tenham sido utilizados para obtenção das informações;
- **Descrição:** breve descrição do jogo avaliado;
- **Forma de utilização:** utilização por usuários individuais ou múltiplos usuários;
- **Tipo do jogo:** identifica se o jogo é utilizado em tabuleiro, cartas, computador em ambiente local ou computador na internet;
- **Área de conhecimento:** área de conhecimento da Engenharia de Software ou área de processo da melhoria de processo de software na qual a atividade de aprendizagem está focada, com base no SWEBOK (IEEE, 2004) e no modelo MPS.BR (SOFTEX, 2011a);
- **Resultados de processo:** resultados de processo que o jogo pode ensinar, com base nos modelo MPS.BR (SOFTEX, 2011a);
- **Atividade de aprendizagem:** visão geral da tarefa que deve ser cumprida pelos alunos;
- **Contexto de uso:** indica o contexto educacional em que o jogo pode ser utilizado;
- **Resultado de aprendizagem:** indica o que o aluno deve atingir como resultado da experiência de aprendizagem, classificados em conhecimentos, habilidades e atitudes;
- **Efeito de aprendizagem:** descrição do efeito de aprendizagem do uso do jogo como ferramenta de ensino; e
- **Principais descobertas:** sumarização dos principais resultados obtidos no estudo e descrição de como o jogo foi avaliado.

3.2 RESULTADOS

Esta seção apresenta os resultados do mapeamento sistemático realizado e está organizada da seguinte forma: a Seção 3.2.1 apresenta como o mapeamento foi efetivamente executado, e a Seção 3.2.3 apresenta a análise dos resultados do mapeamento. As informações extraídas dos trabalhos selecionados são apresentadas no Apêndice A.

3.2.1 Execução

Os trabalhos foram pesquisados do dia 15/10/2010 ao dia 10/11/2010. Visando atualizar este mapeamento, foi realizada uma nova pesquisa do dia 20/10/2012 ao dia 27/10/2012. Para realizar a busca dos trabalhos foram utilizados os termos definidos na Seção 3.1.2.3. A quantidade de trabalhos retornados para cada fonte de pesquisa pode ser conferida na Tabela 9.

Tabela 9. Quantidade de trabalhos retornados por fonte de pesquisa

FONTE DE PESQUISA	QUANTIDADE DE TRABALHOS RETORNADOS
IEEEExplore	280
Portal ACM	88
SpringerLink	51
CiteSeer	25
ScienceDirect	7
WILEY Interscience	36
Google Scholar	19745

Com exceção ao Google Scholar, onde os trabalhos foram conferidos até que se passassem dez páginas sem que um trabalho fosse selecionado (sendo que cada página continha 10 trabalhos), nas demais fontes de pesquisa todos os trabalhos foram conferidos título e resumo levando em consideração os critérios de inclusão e exclusão do protocolo de pesquisa (Seção 3.1.3.1). Considerando todas as fontes de pesquisa e sem desconsiderar os trabalhos duplicados foram selecionados no total 113 trabalhos. A quantidade de trabalhos selecionados por fonte de pesquisa pode ser conferida na Figura 9.

Depois de retirados os trabalhos duplicados, restaram 88 trabalhos que passaram para terceira etapa de seleção. Na terceira etapa de seleção foram conferidos os textos inteiros dos 88 trabalhos selecionados na segunda etapa, utilizando novamente os critérios de inclusão e exclusão. Após a terceira etapa de seleção, restaram 28 trabalhos que tratavam de 12 jogos diferentes.

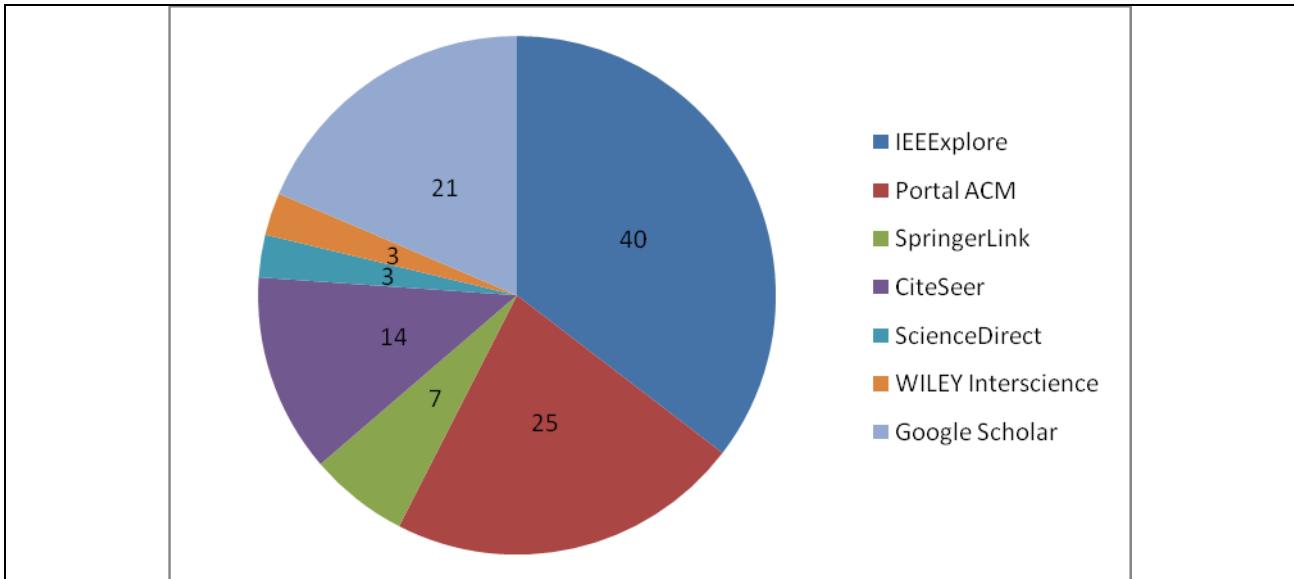


Figura 7. Quantidade trabalhos selecionados por fonte de pesquisa

Além dos 12 jogos selecionados, foram adicionados mais 4 jogos desenvolvidos dentro do grupo de pesquisa que esse trabalho está inserido. Os trabalhos adicionados foram os seguintes: U-TEST (SILVA, 2010), A Ilha dos Requisitos (GONÇALVES, 2010), 123SPI (ADORNO, 2010), e Jogo das 7 Falhas (DINIZ, 2011). Esses jogos foram inseridos, pois atendem a todos os critérios de inclusão e exclusão deste mapeamento, e só não foram selecionados por não possuírem nenhuma publicação relevante. Para consultar os trabalhos desenvolvidos pelo grupo de pesquisa que este trabalho está inserido, podem ser realizadas buscas no currículo do orientador desta pesquisa.

Após a etapa de seleção, os trabalhos selecionados foram analisados e as informações extraídas dos 16 jogos analisados são apresentadas no Apêndice A. A Tabela 10 apresenta os 16 jogos analisados e os resultados esperados do modelo MPS.BR relacionados com cada jogo. Na próxima seção é apresentada a análise dos resultados deste mapeamento sistemático.

Tabela 10. Jogos x resultados esperados

Identificador	Nome do Jogo	Resultados Esperados
1	Groupthink Game	GRE1, GRE2, DRE1, DRE2, DRE3, DRE4, DRE5, DRE6, DRE7, e DRE8
2	MO-SEProcess	GPR3
3	Planager	GPR1 e GPR5
4	PlayScrum	GPR1, GPR3, GPR5, GPR7, GPR8, GPR11, RAP3, RAP4 (Nível G), RAP5 (Até o nível F), RAP8 e RAP10 (Nível G)
5	Problems and Programmers	GPR1, GPR3, GPR7, GPR8, GPR11, RAP3, RAP4 (Nível G), RAP5 (Até o nível F), e RAP10 (Nível G)
6	Re-O-Poly	GRE1, GRE2, GRE5, DRE1, DRE2, DRE3, DRE4, DRE5, DRE6, DRE7, e DRE8
7	SE-RPG	GPR1, GPR3, GPR7, GPR8, RAP5 (Até o nível F), RAP7 (Até o nível F), e RAP10 (Nível G)
8	SimSE	RAP 7, GPR 3, GPR 5, GPR 7, GPR 8 e GPR 11
9	SimulES	GPR 1, GPR 3, GPR 5, GPR 7, GPR 11, GRE 3, RAP 4 (Nível G), RAP 5 (Até o nível F), e RAP10 (Nível G)
10	The Incredible Manager	GPR 1, GPR 2, GPR 4 (Até nível F), GPR 5, GPR 7, GPR 10, GPR 11, GPR 12, GPR 13, RAP 2, RAP 3 (Nível G), RAP 5 (Até nível F), RAP 7 (Até nível F) e RAP 10 (Nível G)
11	X-MED	RAP 4 (A partir do Nível F), RAP 21, RAP 23, RAP 24, RAP 25, RAP 26, RAP 27, RAP 28, RAP 29, RAP 32, RAP 39, RAP 40, RAP 46, MED 1, MED 2, MED 3, MED 4, MED 5, MED 6 e MED 7
12	U-TEST	VER 2 e VAL 2
13	A Ilha dos Requisitos	GRE 1, GRE 2, GRE 3, GRE 4, GRE 5, DRE 1, DRE 2, DRE 3, DRE 4, DRE 6 e DRE 8
14	123SPI	Todos os resultados até o nível C
15	Deliver	GPR 4, GPR 5, GPR 6, GPR 7, GPR 11, GPR 13, GPR 14 e GPR 17
16	Jogo das 7 Falhas	VER 2, VER 4, VER 5, VER 6, VAL 2, VAL 4, VAL 5, VAL 6 e VAL 7

3.2.2 Análise dos Resultados

Em relação à pergunta de pesquisa (Como o conteúdo dos principais modelos de capacidade de software tem sido explorado pelos jogos educacionais existentes?), percebeu-se com os dados extraídos, que o conteúdo dos modelos de capacidade de software é pouco explorado pelos jogos educacionais existentes. Com exceção ao jogo 123SPI (ADORNO, 2012), que visa ajudar a ensinar a implementar os 18 processos do nível 3 do modelo CMMI, não foi encontrado mais nenhum jogo com foco específico no ensino de um modelo de capacidade de software.

Entretanto, foi observado que o conteúdo de todos os demais jogos identificados estão relacionados a pelo menos um dos resultados esperados do modelo MPS.BR. Esses jogos podem ajudar as pessoas a entenderem os conceitos envolvidos nos resultados esperados do MPS.BR. Estes conceitos são necessários para interpretar e implementar o modelo MR-MPS. Portanto, apesar

desses jogos não serem utilizados para demonstrar como implementar estes modelos, eles podem apoiar o entendimento de conceitos e técnicas da Engenharia de Software e da Melhoria de Processo de Software e, como consequência, apoiar uma interpretação mais adequada das práticas e resultados esperados.

Em relação ao efeito de aprendizagem, apesar de boa parte dos jogos não apresentarem uma avaliação formal do efeito de aprendizagem (jogos: 1, 2, 3, 4, 5, 7 e 9), porém a maioria apresenta (jogos: 6, 8, 10, 11, 12, 13, 14, 15 e 16). Isso demonstra que os pesquisadores tem se preocupado em avaliar o efeito aprendizagem da aplicação dos jogos como ferramenta de ensino. Dentre os jogos que apresentaram uma avaliação formal do efeito de aprendizagem, em dois deles foi realizada somente uma análise qualitativa (jogos 10 e 15), os demais jogos abordaram tanto análise qualitativa quanto quantitativa. De acordo com análise qualitativa, em todos os jogos analisados os alunos apontaram que os jogos são ferramentas motivadoras para o ensino. Dos jogos que realizaram análise quantitativa, somente em dois deles não foi possível comprovar que o jogo apresentou um efeito de aprendizagem positivo (jogos: 11 e 14).

3.3 DIFERENCIAL DO TRABALHO

Com base na análise isolada dos jogos selecionados no mapeamento sistemático, foi possível verificar algumas características comuns e diferenças entre esses jogos e o jogo desenvolvido neste trabalho. A seguir essas características são apresentadas.

O jogo 123SPI (ADORNO, 2012) possui objetivos similares aos do jogo desenvolvido neste trabalho, pois ambos visam apoiar o ensino de em um modelo de capacidade de software. Porém, enquanto o jogo 123SPI é focado no modelo CMMI níveis 2 e 3 de maturidade, o jogo desenvolvido neste trabalho é focado no modelo MPS.BR nível G. Apesar dos demais jogos poderem ser utilizados para ajudar as pessoas a entenderem os conceitos relacionados com os modelos de capacidade de software, porém eles não relacionam esses conceitos com as práticas e resultados esperados desses modelos. No jogo desenvolvido neste trabalho as situações vivenciadas durante o jogo são relacionados aos resultados esperados do MPS.BR, permitindo que os jogadores saibam quais práticas foram discutidas em cada situação do jogo.

Assim como nos jogos SE-RPG (BENITTI & MOLLÉRI, 2008) e SimSE (NAVARRO, 2006), o jogo desenvolvido neste trabalho simula uma empresa de desenvolvimento de software,

onde o jogador pode contratar e demitir desenvolvedores, acompanhar o andamento dos projetos, consultar a produtividade dos desenvolvedores, etc. Porém, ao contrário desses jogos, o objetivo principal do jogo desenvolvido não está em completar os projetos com sucesso. O objetivo principal do jogo desenvolvido está em resolver as situações que são apresentadas durante o jogo e atender os resultados esperados do MPS.BR.

Ao comparar os jogos analisados em relação ao aspecto lúdico, o jogo que mais se destaca é A Ilha dos Requisitos (GONÇALVES, 2010). A história desse jogo se passa em uma ilha deserta que possui um vulcão que está prestes a explodir, e o jogador no papel de um analista de requisitos vai parar nesta ilha após sofrer um acidente de avião. Esta ilha também é habitada por canibais que acham que o analista de requisitos é um enviado de seu Deus para salvá-los da grande explosão do vulcão que está prestes a ocorrer. Para salvar a si próprio e todos os habitantes da ilha desta explosão, o analista de requisitos deve então conseguir resolver uma série de desafios relacionados com o processo de Engenharia de Requisitos. Já a história do jogo desenvolvido neste trabalho se passa em uma cidade conhecida pela grande quantidade de empresas de desenvolvimento de software de alta qualidade, na qual o jogador terá recebido um investimento para montar sua própria empresa nesta cidade, mas como restrição ele deve alcançar o nível G do MPS.BR no prazo máximo de um ano.

Em relação a como o conteúdo é ensinado durante o jogo, alguns jogos realizam perguntas objetivas que os jogadores devem responder para poderem avançar no jogo (SMITH & GOTEL, 2008; WANGENHEIM *et al.*, 2009; SILVA, 2010; GOLÇALVES, 2010; ADORNO, 2012), outros simulam a execução de atividades do dia a dia de um profissional da computação (YE, LIU & POLACK-WAHL, 2007; PRIKLADNICKI, ROSA, KIELING, 2007; DINIZ, 2011), mas a maioria dos jogos encontrados simulam a execução de um projeto de software em que o jogador deve tomar decisões para conseguir concluir o projeto com sucesso (ZHU, WANG & TAN, 2007; FERNANDES & SOUZA, 2010; BAKER, NAVARRO & HOEK, 2005; BENITTI & MOLLÉRI, 2008; NAVARRO, 2006; FIGUEIREDO *et al.* 2007; DANTAS, BARROS & WERNER, 2004; WANGENHEIM, SAVI & BORGATTO, 2011). No jogo desenvolvido neste trabalho o conteúdo é ensinado por meio da simulação de situações do dia a dia de empresas de desenvolvimento de software em que com os resultados esperados do MPS.BR são indiretamente discutidos.

4 DESENVOLVIMENTO

Neste capítulo o jogo desenvolvido neste trabalho é apresentado. Inicialmente são apresentados os objetivos educacionais, em seguida é apresentado o Game Design, que inclui dois artefatos, o *Game Design Document* (GDD) e o *Technical Design Document* (TDD). Posteriormente, são apresentados os eventos do jogo, que são situações relacionadas aos resultados esperados do MPS.BR que o jogador terá que resolver. E por fim o jogo desenvolvido é apresentado.

4.1 OBJETIVOS EDUCACIONAIS

O público alvo do jogo são alunos de graduação, pós-graduação e profissionais da área de Computação/Informática, com conhecimentos básicos em Engenharia de Software e melhoria de processo de software. Os conhecimentos básicos em Engenharia de Software permitem ao aluno compreender o processo de desenvolvimento de software, a importância de alguns processos de software e seus diferentes momentos de aplicação em um projeto de software. Enquanto que os conhecimentos básicos em melhoria de processo de software permitem ao aluno conhecer os diferentes modelos de melhoria de processo e compreender os conteúdos explorados pelo jogo.

O jogo visa apoiar o ensino de melhoria de processo de software, com foco no modelo de Melhoria do Processo de Software Brasileiro – MPS.BR (SOFTEX, 2011a). Inicialmente o jogo será focado somente nos processos do nível G de maturidade deste modelo. Os objetivos educacionais definidos a seguir possuem como base a taxonomia revisada de Bloom (ANDERSON & KRATHWOHL, 2001), e compreendem os níveis Lembrar, Entender e Aplicar. Assim, são objetivos educacionais do jogo:

1. Lembrar e compreender os resultados esperados do processo Gerência de Requisitos do nível G de maturidade do MPS.BR;
2. Lembrar e compreender os resultados esperados do processo Gerência de Projetos do nível G de maturidade do MPS.BR;
3. Aplicar os resultados esperados do processo Gerência de Requisitos do nível G de maturidade do MPS.BR; e

4. Aplicar os resultados esperados do processo Gerência de Projetos do nível G de maturidade do MPS.BR.

4.2 GAME DESIGN

O Game Design geralmente envolve a geração de dois artefatos conhecidos como *Game Design Document* (GDD) e *Technical Design Document* (TDD) (RABOWSKY, 2009). As seções 4.2.1 e 4.2.2 apresentam respectivamente o GDD e o TDD do jogo desenvolvido neste trabalho.

4.2.1 Game Design Document (GDD)

O GDD descreve os itens relacionados a jogabilidade do jogo. O modelo de GDD utilizado neste trabalho foi baseado em Azevedo (2008) e no *framework* para desenvolvimento de jogos educacionais (YUSOFF *et al.*, 2009) apresentado na Seção 2.2.5 .

1 Resumo do Projeto

1.1 Conceito do Jogo

SPI City é um jogo educacional para apoio à capacitação em melhoria de processo de software, com foco específico no nível G de maturidade do MPS.BR. O objetivo do jogo é ajudar os jogadores a lembrarem, compreenderem e aplicarem os resultados esperados do modelo MPS.BR.

1.2 Conjunto de Características

Jogo dinâmico com gráficos e animações em 2D, desenvolvido em Flash e executado via web.

1.3 Gênero

SPI City é um jogo educacional no estilo simulador, pois reproduz situações do dia a dia de empresas de desenvolvimento de software.

1.4 PÚBLICO-ALVO

Alunos de graduação, pós-graduação e profissionais da área de Computação/Informática.

1.5 Resumo do Fluxo do Jogo

Inicialmente o jogador deverá se cadastrar no jogo, informando seus dados na tela de cadastro. Após o cadastro o jogador poderá fazer login no jogo, utilizando o nome de usuário e senha informados no momento do cadastro. Após o login o jogador deverá criar sua empresa para poder iniciar o jogo. Ao iniciar o jogo o jogador terá como cenário principal o ambiente de uma empresa de desenvolvimento de software, e o jogador no papel de gerente e dono da empresa poderá contratar, demitir, treinar, e recompensar funcionários, acompanhar o andamento dos projetos, entre outras coisas. Porém, a principal tarefa do jogador é resolver as situações apresentadas durante o jogo. Essas situações estão relacionadas com os resultados esperados do nível G de maturidade do MPS.BR, e a medida que o jogador for resolvendo essas situações, ele atenderá os resultados esperados associados com cada situação. O jogo será completado quando o jogador conseguir resolver todas as situações disponíveis. O jogo será finalizado caso o dinheiro da empresa fique negativo ou, caso passe um ano no tempo do jogo e o jogador ainda não conseguiu resolver todas as situações disponíveis.

1.6 Escopo do Projeto

1.6.1 Número de Cenários

O jogador terá como cenário principal o ambiente de uma empresa de desenvolvimento de software, dividida em três áreas:

- Setor de desenvolvimento: setor onde os projetos da empresa são desenvolvidos, neste setor o jogador poderá contratar, demitir, treinar, e recompensar desenvolvedores, poderá consultar o andamento dos projetos, realizar reuniões de acompanhamento, e poderá conversar com os desenvolvedores ou com o gerente de projetos quando eles estiverem com alguma dúvida;
- Setor administrativo: neste setor o jogador poderá consultar suas tarefas, através da secretaria, poderá consultar os gastos e receitas de sua empresa, através do contador, e poderá consultar os resultados esperados atendidos, através do implementador; e
- Sala do consultor: nesta sala o jogador poderá conversar com o consultor Humphrey para tirar suas dúvidas sobre conceitos de melhoria de processo de software.

1.6.2 Níveis do Jogo

Os níveis do jogo corresponderão aos níveis de maturidade do modelo MPS.BR (SOFTEX, 2011a). Porém inicialmente o jogo possuirá somente um nível, o nível G. Para alcançar este nível os jogadores deverão responder corretamente todos os diálogos do jogo, até que se consiga implementar todos os resultados esperados dos processos Gerência de Requisitos e Gerência de Projetos do nível G de maturidade do MPS.BR.

2 Jogabilidade e Mecânica

2.1 Jogabilidade

2.1.1 Progressão do Jogo

Ao acessar o jogo será exibida uma animação que apresenta a cidade de SPI City e conta que o jogador recebeu um investimento para montar sua empresa, mas como exigência ele deverá alcançar o nível G de maturidade do MPS.BR no prazo máximo de um ano. Após a apresentação da animação, ou caso o jogador tenha clicado em pular introdução, será apresentada a tela de login. Nesta tela o jogador deverá informar seu nome de usuário e sua senha para poder entrar no jogo, ou clicar em cadastrar caso ele ainda não possua cadastro. Para se cadastrar no jogo o jogador deverá informar os seguintes dados: nome, email, instituição/empresa, nome de usuário e senha. Após realizar o login, será exibido o menu principal do jogo composto pelas seguintes opções: empresas, tutorial, sobre, regras, ranking de empresas e sair. No menu principal o jogador deve acessar a opção “empresas” para visualizar suas empresas. Na tela de empresas o jogador deve clicar em “Criar empresa”, caso ele ainda não possua uma empresa cadastrada ou deseja criar uma nova. Para cadastrar uma empresa o jogador deve informar os seguintes dados: nome da empresa, nome do produto, linguagem de programação utilizada e plataforma de desenvolvimento (web ou desktop). Para iniciar um jogo, o jogador deve selecionar uma empresa cadastrada e clicar em “Jogar.”

Ao iniciar um jogo será exibido o interior de uma empresa de desenvolvimento de software dividida em três áreas: setor administrativo, setor de desenvolvimento, e sala do consultor. A primeira área exibida é a do setor administrativo. O jogador poderá navegar entre as áreas da empresa clicando nos botões localizados na parte inferior direita da tela. No setor administrativo o jogador poderá: consultar suas tarefas para obter informações sobre o que ele deve fazer naquele

momento, poderá obter informações sobre os gastos e receitas de cada mês, e poderá obter informações sobre quais resultados esperados já foram atendidos. No setor de desenvolvimento os projetos da empresa são desenvolvidos. Neste setor o jogador poderá: contratar, demitir, treinar, e recompensar desenvolvedores; consultar o andamento dos projetos; realizar reuniões de acompanhamento; e conversar com os desenvolvedores ou com o gerente de projetos quando eles estiverem com alguma dúvida. Na sala do consultor o jogador poderá tirar suas dúvidas sobre conceitos de melhoria de processo de software.

A principal tarefa do jogador será resolver determinadas situações, relacionadas com os resultados esperados do MPS.BR, que serão apresentadas durante o jogo. A medida que o jogador for resolvendo essas situações, ele atenderá os resultados esperados associados com cada situação. Para resolver uma situação o jogador deve conversar com o personagem associado com a situação, e responder corretamente todas as perguntas feitas pelo personagem sobre aquela situação. O jogo será completado quando o jogador conseguir resolver todas as situações disponíveis, e será finalizado com fracasso caso o dinheiro da empresa fique negativo, ou caso passe um ano no tempo do jogo e o jogador ainda não conseguiu resolver todas as situações disponíveis.

2.1.2 Fluxo do Jogo

A jogabilidade proposta para o jogo pode ser melhor entendida observando o fluxo do jogo apresentado na Figura 8.

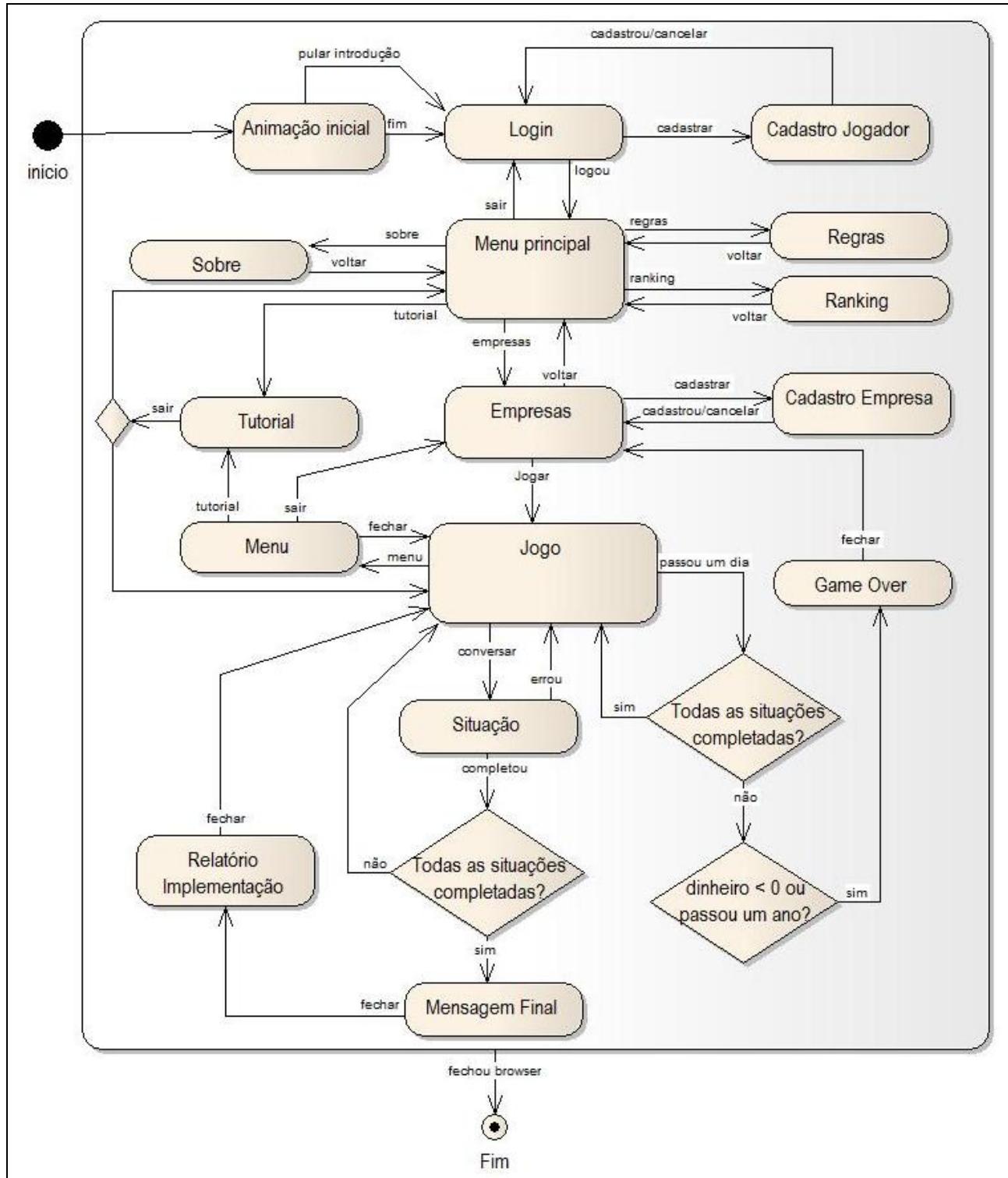


Figura 8. Fluxo do jogo

2.1.3 Opções do Jogo

Cada diálogo no jogo possui uma ou mais opções de respostas, porém para a primeira versão do jogo só haverá uma opção correta, assim a progressão do jogo não será afetada pelas escolhas dos jogadores nos diálogos.

2.1.4 Re-jogando e Salvando o Jogo

Para a primeira versão, o progresso do jogo não será salvo, ou seja, se o jogador sair do jogo ele perde o seu progresso e deverá executar o jogo a partir do início novamente. Porém o jogador pode pausar e continuar o jogo sempre que desejar, desde que ele não feche a janela do jogo.

2.2 Mecânica

2.2.1 Setor de desenvolvimento

No setor de desenvolvimento o jogador poderá: contratar, demitir, treinar, e recompensar desenvolvedores; consultar o andamento dos projetos; realizar reuniões de acompanhamento; e conversar com os desenvolvedores ou com o gerente de projetos quando eles estiverem com alguma dúvida. A seguir é apresentado como cada uma destas atividades serão executadas no jogo.

2.2.1.1 Contratar e demitir

Inicialmente a empresa não possuirá nenhum programador e nenhum analista, mas possuirá um gerente de projetos. Para poder iniciar o desenvolvimento de seu produto o jogador precisa contratar pelo menos um programador e um analista para compor sua equipe de desenvolvimento. Para contratar um programador ou um analista, o jogador deve clicar sobre uma mesa vazia e selecionar a opção “Contratar”, depois selecionar se ele quer contratar um analista ou um programador. Feito isto o funcionário contratado passa a utilizar a mesa em que o jogador selecionou.

Para demitir um programador ou um analista o jogador deve clicar sobre o mesmo e selecionar a opção “Demitir”. Feito isto a mesa onde estava o programador ou analista passa a ficar vazia. Ao demitir um funcionário será descontado R\$ 3000,00 do dinheiro da empresa referente aos

encargos salariais. O gerente de projeto é um personagem auxiliar no jogo e por isso não poderá ser demitido pelo jogador.

2.2.1.2 Treinar

Para treinar um programador ou um analista o jogador deve clicar sobre o mesmo, selecionar a opção “Treinar” e depois selecionar o tipo de treinamento que será realizado. Os tipos de treinamento disponíveis são: processo, gerência de projetos, gerência de requisitos e tecnologia de programação. O treinamento não terá um custo financeiro para empresa, porém o funcionário que está sendo treinado ficará dois dias sem produzir. Após o treinamento a habilidade do funcionário será incrementada (ver Item 2.2.5.2 do GDD) e consequentemente também a sua produtividade.

O objetivo do treinamento no jogo é permitir que o jogador entenda a importância de seus colaboradores serem treinados, pois segundo Wangenheim, Hauck e Wangenheim (2009a), o treinamento é fundamental para motivar as pessoas e fornecer suporte para que elas desenvolvam as capacidades necessárias para executar suas tarefas. E ainda no resultado esperado GPR7 do MPS.BR, espera-se que as pessoas sejam treinadas quando elas não possuírem as competências necessárias para executar suas tarefas.

2.2.1.3 Recompensar

Outro fator que afeta a produtividade de um funcionário é a motivação do mesmo. Caso um programador ou analista venha apresentando baixa motivação o jogador pode fornecer uma recompensa para tentar aumentar sua motivação e consequentemente a sua produtividade. Para fornecer uma recompensa para um programador ou analista, o jogador deve clicar sobre o mesmo e selecionar a opção “Recompensar”. O custo da recompensa para a empresa será de R\$ 500,00 e a motivação do funcionário recompensado irá aumentar conforme apresentado na Tabela 11.

O objetivo da recompensa no jogo é permitir que o jogador tenha outras maneiras de aumentar a produtividade dos seus colaboradores. Porém a recompensa não é a melhor maneira de aumentar a produtividade dos colaboradores, pois se a produtividade está baixa é por que está faltando treinamento. A recompensa até pode ajudar aumentar a produtividade momentaneamente, mas se o jogador não treinar seus colaboradores a produtividade irá baixar novamente. Por isso o

valor da recompensa é relativamente alto, para evitar que o jogador recompense seus colaboradores frequentemente.

2.2.1.4 Consultar informações sobre os desenvolvedores

Para consultar informações sobre um programador ou analista como função, motivação, habilidade e produtividade, o jogador deve ir com o mouse sobre o mesmo e as informações aparecerão na barra inferior do jogo.

2.2.1.5 Consultar status do projeto

Para consultar o status do projeto o jogador deve clicar sobre o gerente de projetos e selecionar a opção “Status do projeto”. Feito isto aparecerá uma janela com informações sobre o projeto como: status (atrasado ou em dia), porcentagem de conclusão, data de início, data de término prevista, progresso, tamanho estimado, custo estimado, etc.

2.2.1.6 Realizar reuniões de acompanhamento

O gerente de projetos também é responsável por fazer as reuniões de acompanhamento com a equipe de desenvolvimento. As reuniões de acompanhamento são importantes para manter os membros da equipe atualizados sobre a situação dos projetos e para manter o comprometimento dos mesmos com o projeto, conforme estabelece o resultado esperado “GPR12 - O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido” (SOFTEX, 2011c).

Para que uma reunião de acompanhamento seja realizada o jogador deve clicar sobre o gerente de projetos e selecionar a opção “Realizar reunião de acompanhamento”. Feito isto, o jogador receberá um *feedback* mostrando que a reunião de acompanhamento foi realizada e a motivação dos desenvolvedores será incrementada conforme apresentado na Tabela 11.

2.2.1.7 Conversar

O jogador deverá conversar com os membros da equipe de desenvolvimento ou com o gerente de projetos, quando eles precisarem da ajuda do jogador para resolver alguma situação.

Cada situação será uma tarefa que o jogador deverá realizar no jogo (veja Apêndice B para consultar as situações disponíveis no jogo). Para conversar com um personagem o jogador deverá clicar sobre o personagem e selecionar a opção “Conversar”. Feito isso irá aparecer uma tela simulando um diálogo entre o jogador e o personagem que o jogador selecionou. A opção “Conversar” só estará disponível para um personagem quando houver uma situação para ser resolvida relacionada com este personagem.

2.2.2 Setor administrativo

No setor administrativo o jogador poderá: consultar suas tarefas; consultar o relatório de gastos e receitas da empresa; e consultar o relatório de implementação. A seguir é apresentado como cada uma dessas atividades pode ser realizada no jogo.

2.2.2.1 Visualizar tarefas

Cada situação será uma tarefa que o jogador deverá realizar no jogo (veja Apêndice B para consultar as situações disponíveis no jogo). Para consultar suas tarefas o jogador deve clicar sobre a secretaria e no menu que aparecer selecionar a opção “Visualizar tarefas”. Feito isso será exibida uma tela com a lista de tarefas que o jogador já executou e com a tarefa que ainda não foi realizada, caso haja alguma. As tarefas que já foram executadas estarão marcadas com uma bolinha verde e a que ainda não foi executada estará marcada com uma bolinha vermelha. Clicando sobre uma tarefa o jogador visualizará informações sobre o que deve ser realizado na tarefa.

2.2.2.2 Consultar relatório de gastos e receitas

Para consultar o relatório de gastos e receitas de cada mês o jogador deverá clicar sobre o contador e no menu que aparecer selecionar a opção “Relatório de gastos e receitas”. Feito isso será exibida uma tela contendo uma lista com todos os meses do ano, e o jogador deverá clicar sobre o mês na qual ele deseja visualizar os gastos e receitas.

2.2.2.3 Consultar relatório de implementação

Para obter informações sobre quais resultados esperados já foram atendidos, o jogador deverá clicar sobre o implementador e no menu que aparecer selecionar a opção “Relatório de implementação”. Feito isso será exibida uma tela contendo uma lista com as siglas de todos os resultados esperados do nível G de maturidade do MPS.BR. Clicando sobre um resultado esperado o jogador poderá visualizar o status deste resultado (totalmente implementado, largamente implementado, parcialmente implementado, ou não avaliado), a descrição do resultado esperado, e os diálogos realizados até o momento associados com o resultado esperado.

2.2.3 Sala do consultor

Nesta sala o jogador poderá conversar com o consultor Humphrey para tirar suas dúvidas sobre melhoria de processo de software. O jogador irá conversar com o consultor através de um chat, e para realizar uma pergunta para o consultor o jogador deve inserir a pergunta no campo de entrada e clicar em enviar. Feito isso, a pergunta feita pelo jogador e a resposta do consultor serão inseridas no campo de texto que guarda a conversa entre o jogador e o consultor, e o jogador poderá visualizar a resposta recebida.

O consultor Humphrey é um chatterbot que faz parte de outra pesquisa em desenvolvimento, por isso não serão apresentados detalhes de sua implementação. O consultor Humphrey é um mecanismo que foi adicionado ao jogo para permitir os jogadores esclarecerem suas dúvidas sobre conceitos de Engenharia de Software e melhoria de processos, porém a sua base de conhecimentos ainda possui poucos conceitos de melhoria de processo.

2.2.4 Controle de tempo

O tempo do jogo é controlado por um relógio fictício que marca o dia, o mês e o ano em que o jogo se encontra. Ao começar um novo jogo o relógio começa a contar a partir do dia 1 de janeiro de 2012. A cada 30 segundos do tempo real um dia é incrementado no relógio fictício do jogo. O relógio fictício deve permitir o usuário pausar e continuar o jogo. Ao pausar o jogo o relógio para de ser incrementado e tudo no jogo fica pausado, e o jogador poderá somente navegar entre os

ambientes e acessar o menu do jogo, não podendo executar mais nenhuma ação no jogo até que ele clique em continuar.

2.2.5 Variáveis do jogo

Esta seção apresenta algumas variáveis do jogo e como elas serão calculadas, e está organizada da seguinte forma: a Seção 2.2.5.1 apresenta a motivação dos desenvolvedores e como ela será calculada, a Seção 2.2.5.2 apresenta a habilidade dos desenvolvedores e como ela será calculada, a Seção 2.2.5.3 apresenta a produtividade dos desenvolvedores e como ela será calculada, e a Seção 2.2.5.4 apresenta o dinheiro da empresa e as situações que farão o dinheiro aumentar ou diminuir.

2.2.5.1 Motivação

A motivação de um membro da equipe de desenvolvimento será utilizada no cálculo da produtividade do mesmo e será um valor entre 0 e 100. Quando um membro da equipe de desenvolvimento é contratado o valor da sua motivação será 100. A motivação pode baixar ou aumentar de acordo com algumas situações do jogo, como apresentado na Tabela 11. De acordo com a situação ocorrida e do valor da motivação atual do desenvolvedor, o valor de sua motivação pode baixar ou aumentar mais. Por exemplo: se o projeto está atrasado e o valor da motivação do desenvolvedor está entre 80 e 100, o valor que será diminuído da motivação do mesmo será um valor aleatório entre 15 e 20.

Tabela 11. Situações que farão a motivação diminuir ou aumentar

Situação	Quando Avaliar	Motivação	Variação
Membro da equipe de desenvolvimento não recebeu treinamento no processo da empresa	Na primeira, na segunda e na terceira semana após a contratação	(80,100]	- [20,30]
		(70,80]	- [10,20]
		(60,70]	- [5,10]
		(50,60]	- [1,5]
Projeto atrasado	Diariamente	(80,100]	- [15,20]
		(70,80]	- [10,15]
		(60,70]	- [5,10]
		(50,60]	- [1,5]
Não foi feito nenhuma reunião de acompanhamento no projeto	Ao final do projeto	(80,100]	- [10,15]
		(60,80]	- [5,10]
		(40,60]	- [1,5]
Membro da equipe de desenvolvimento recebeu recompensa	Quando ocorrer	[0,50]	+ [20,30]
		(50,70]	+ [10,20]
		(70,90]	+ [5,10]
		(90,95]	+ [1,5]
Projeto entregue no prazo	Quando ocorrer	[0,50]	+ [15,25]
		(50,70]	+ [10,20]
		(80,90]	+ [5,10]
		(90,95]	+ [1,5]
Foi realizada uma reunião de acompanhamento	Quando ocorrer	[0,50]	+ [15,25]
		(50,70]	+ [10,20]
		(80,90]	+ [5,10]
		(90,95]	+ [1,5]

2.2.5.2 Habilidade

A habilidade de um membro da equipe de desenvolvimento será utilizada no cálculo da produtividade do mesmo e será um valor entre 0 e 100. O valor inicial da habilidade de um desenvolvedor será um valor aleatório entre 20 e 60 e será calculado no momento da sua contratação. O valor da habilidade de um membro da equipe de desenvolvimento pode subir se o mesmo for treinado de acordo com as seguintes regras:

- Se um analista ou programador for treinado no processo da empresa o valor da sua habilidade será incrementada em 20;
- Se um programador for treinado em tecnologia de desenvolvimento o valor da sua habilidade será incrementado em 20; e
- Se um analista for treinado em gerência de requisitos o valor da sua produtividade será incrementado em 20.

2.2.5.3 Produtividade

A produtividade de um membro da equipe de desenvolvimento será utilizada para calcular o andamento dos projetos e será um valor entre 0 e 100. O cálculo da produtividade de um membro da equipe de desenvolvimento será realizado diariamente e envolverá três variáveis que são:

- Tempo de empresa: tempo de empresa em dias de um membro da equipe de desenvolvimento. O valor relacionado ao tempo de empresa que será utilizado no cálculo da produtividade será obtido da seguinte forma:
 - Se o tempo de empresa for menor que 100 dias, o valor que será utilizado no cálculo da produtividade será 25;
 - Se o tempo de empresa for maior ou igual a 100 e menor que 200 dias, o valor que será utilizado no cálculo da produtividade será 50;
 - Se o tempo de empresa for maior ou igual a 200 e menor que 300 dias, o valor que será utilizado no cálculo da produtividade será 75; e
 - Se o tempo de empresa for maior ou igual a 300 dias, o valor que será utilizado no cálculo da produtividade será 100.
- Motivação: valor da motivação de um membro da equipe de desenvolvimento; e
- Habilidade: valor da habilidade de um membro da equipe de desenvolvimento.

O valor da produtividade de um membro da equipe de desenvolvimento será a soma dos valores das três variáveis apresentadas anteriormente divididas por três, de acordo com a seguinte fórmula:

- $\langle \text{produtividade} \rangle = (\langle \text{valor tempo empresa} \rangle + \langle \text{motivação} \rangle + \langle \text{habilidade} \rangle) / 3$.

2.2.5.4 Dinheiro

Ao iniciar o jogo a quantidade de dinheiro inicial da empresa será R\$ 50000,00. A cada final do mês será descontado do dinheiro da empresa um valor relacionado ao pagamento dos funcionários e ao custo da infra-estrutura da empresa. O valor dos salários dos funcionários de

acordo com o seu papel é apresentado na Tabela 12, e o custo de infra-estrutura será R\$ 200,00 por funcionário. Assim, o custo mensal da empresa relacionado ao pagamento dos funcionários e o custo de infra-estrutura, será a soma dos salários dos funcionários mais o custo da infra-estrutura multiplicado pela quantidade de funcionários, de acordo com a seguinte fórmula:

- $\text{<custoMensalEmpresa>} = \sum \text{<salarioFuncionario>} + (\text{<custoInfraestutura>} * \text{<quantidadeFuncionario>})$

Se um funcionário for demitido será descontado do dinheiro da empresa R\$ 3000,00 referente aos encargos salariais, e se um funcionário for recompensado será descontado R\$ 500,00 referente ao pagamento da recompensa.

Tabela 12. Salário dos funcionários de acordo com seu papel

Funcionário	Salário
Secretária	R\$ 500,00
Contador	R\$ 800,00
Programador	R\$ 1000,00
Analista	R\$ 1200,00
Gerente de Projetos	R\$ 1500,00

A empresa ganhará dinheiro com a venda de seu produto, porém ela só começará a vender seu produto a partir da finalização do seu primeiro projeto. Inicialmente o produto será vendido por R\$ 1000,00 cada cópia e a cada evolução o seu preço aumentará em R\$ 200,00. Os clientes que já tiverem o produto terão que pagar R\$ 200,00 para terem direito a nova versão.

A primeira versão do produto começará vendendo 10 cópias por mês e a cada nova versão o número de cópias vendidas por mês é incrementado em 5 cópias, caso o projeto da nova versão não tenha atrasado mais que 5 dias. Caso o projeto tenha atrasado mais que 5 dias, o número de cópias vendidas é decrementado em 2 até alcançar o limite de 2 cópias por mês.

O número de clientes que já possuem o produto e que irão querer a nova versão será igual a quantidade de clientes da empresa. Porém, se o projeto tenha atrasado mais que 5 dias a quantidade de clientes da empresa cai pela metade, ou seja, somente metade dos clientes que já possuem o produto irão querer a nova versão.

2.2.6 Projetos

O produto da empresa será evoluído através de projetos e cada projeto está associado com uma versão do produto. A empresa desenvolverá somente um projeto por vez e assim que um projeto termina um novo é imediatamente iniciado. Esta seção apresenta as variáveis dos projetos e como elas serão calculadas, e está organizada da seguinte forma: a Seção 2.2.6.1 apresenta o tamanho dos projetos, a Seção 2.2.6.2 apresenta o esforço, a Seção 2.2.6.3 apresenta o custo e a Seção 2.2.6.4 apresenta o progresso dos projetos.

2.2.6.1 Tamanho

O tamanho real de um projeto será um valor fictício aleatório entre 1000 e 3000 e será determinado no momento em que o planejamento do mesmo é iniciado. O tamanho real não deve aparecer para o jogador, ele será utilizado para calcular o progresso dos projetos e servirá de base para o cálculo das demais estimativas do projeto.

O tamanho estimado de um projeto será calculado de acordo com uma taxa de variação em relação ao tamanho real do projeto. A taxa de variação será um valor aleatório entre -20 e 20%. O valor da estimativa de tamanho será calculado somando o valor da taxa variação com 1 e multiplicando o resultado pelo tamanho real do projeto, de acordo com a seguinte fórmula:

- $\langle \text{tamanhoEstimado} \rangle = \langle \text{tamanhoReal} \rangle * (\langle \text{taxaVariacao} \rangle + 1)$

2.2.6.2 Esforço

A estimativa de esforço de um projeto determina a duração do mesmo em dias, e será calculado dividindo o tamanho estimado do projeto pela produtividade da equipe, de acordo com a seguinte fórmula:

- $\langle \text{esforcoEstimado} \rangle = \langle \text{tamanhoEstimado} \rangle / \langle \text{produtividadeEquipe} \rangle$

A produtividade da equipe será calculada somando-se o valor da produtividade de cada membro da equipe de desenvolvimento, de acordo com a seguinte fórmula:

- $\langle \text{produtividadeEquipe} \rangle = \sum \langle \text{produtividade} \rangle$

2.2.6.3 Custo

A estimativa de custo de um projeto determina quanto o projeto custará em dinheiro para ser desenvolvido. A estimativa de custo será calculada dividindo o custo mensal da empresa por 30 e multiplicando o resultado pela estimativa de esforço do mesmo, de acordo com a seguinte fórmula:

- $\langle \text{custoEstimado} \rangle = \langle \text{custoMensalEmpresa} \rangle / 30 * \langle \text{esforcoEstimado} \rangle$

2.2.6.4 Progresso

O progresso é utilizado para controlar o andamento de um projeto e para determinar quando um projeto termina. Quando o progresso alcançar o tamanho real, o projeto termina. O progresso do projeto será calculado somando o progresso do dia anterior com a produtividade da equipe do dia. A fórmula para o cálculo do progresso é a seguinte:

- $\langle \text{progresso} \rangle = \langle \text{progresso} \rangle + \langle \text{produtividadeEquipe} \rangle$

Para saber se o projeto está atrasado será utilizado o índice de desempenho de prazos (SPI – *Schedule Performance Index*) da técnica de análise de valor agregado (EVA – *Earned Value Analysis*) (PMI, 2008). O índice de desempenho de prazos é uma medida do progresso alcançado comparado com progresso planejado em um projeto, através dele é possível saber se o projeto está atrasado ou adiantado e quantos por centos o projeto está atrasado ou adiantado. O SPI é calculado dividindo o progresso real do projeto pelo progresso planejado, de acordo com a seguinte fórmula:

- $\langle \text{SPI} \rangle = \langle \text{progresso} \rangle / \langle \text{progressoPlanejado} \rangle$

Para calcular o progresso planejado é só dividir o tamanho real do projeto pelo seu esforço estimado e depois multiplicar pela quantidade de dias decorridos em um projeto até o momento, de acordo com a seguinte fórmula:

- $$\langle \text{progressoPlanejado} \rangle = \langle \text{tamanhoReal} \rangle / \langle \text{esforcoEstimado} \rangle * \langle \text{quantidadeDiasDecorridos} \rangle$$

Se o valor do SPI for menor que 1 significa que o projeto está atrasado em relação ao seu cronograma e se for maior que 1 significa que o projeto está adiantado em relação ao seu cronograma. Para saber a porcentagem que um projeto está atrasado ou adiantado é só diminuir o SPI por 1, o resultado deve ser colocado em módulo e depois multiplicado por 100, de acordo com a seguinte fórmula:

- $\text{<porcentagem>} = |\text{<SPI>} - 1| * 100$

2.2.7 Situações

A principal tarefa do jogador será resolver determinadas situações que serão apresentadas durante o jogo. Essas situações serão apresentadas quando as condições para que elas sejam apresentadas sejam satisfeitas. Cada situação possui um personagem associado e uma lista de diálogos. Para resolver uma situação o jogador deve conversar com o personagem associado com a situação e responder corretamente todos os diálogos. Os diálogos são perguntas que o jogador deve responder simulando uma conversa entre o jogador e o personagem associado com a situação. Cada diálogo possui uma ou mais opções de resposta e o jogador pode selecionar somente uma opção como resposta para um diálogo, existindo somente uma opção correta.

Ao selecionar uma opção de resposta o jogador recebe um feedback do implementador indicando se a opção escolhida está correta ou incorreta, e explicando a opção escolhida ou trazendo alguma informação adicional sobre o diálogo. Se o jogador escolher uma opção incorreta a conversa com o personagem é finalizada e o jogador deve aguardar um dia para voltar a conversar com aquele personagem e continuar a resolver a situação. Ao continuar uma situação, ela é retomada a partir do diálogo em que o jogador selecionou a opção incorreta. Se o jogador selecionar a opção correta será apresentado o próximo diálogo, até que se chegue ao último diálogo da situação. Após responder corretamente o último diálogo, a tarefa relacionada àquela situação é concluída.

Cada diálogo possui um ou mais resultados esperados associados e a indicação do status do resultado esperado. Ao responder corretamente um diálogo, os status dos resultados esperados associados serão alterados para os status indicados neste diálogo. Além disso, a pergunta, a resposta e o feedback recebido neste diálogo serão copiados para o log de cada resultado esperado associado, para que os diálogos associados com cada resultado esperado sejam apresentados no relatório de implementação. Ao final de uma situação, os resultados esperados atendidos serão apresentados,

onde um resultado esperado é considerado como atendido se seu status for igual a T (totalmente implementado).

2.2.8 Ranking

Para cada diálogo respondido corretamente o jogador ganhará pontos que servirão para montar o ranking do jogo. Para calcular a pontuação é considerado o tempo de resposta e a quantidade de tentativas para responder corretamente um diálogo. Se o jogador responder corretamente um diálogo na sua primeira tentativa e o tempo que ele levou para responder for abaixo de 60 segundos, ele receberá 100 pontos. Se o jogador responder corretamente um diálogo na sua primeira tentativa, mas ele levou mais de 60 segundos para responder, ele receberá somente 50 pontos. Se jogador não responder corretamente na primeira tentativa, mas sim na segunda tentativa, ele receberá 40 pontos, ou seja, para cada vez que o jogador não responder corretamente um diálogo, ele não poderá mais receber 100 pontos para aquele diálogo e serão descontados 10 pontos para cada resposta incorreta, até que ele atinja a pontuação mínima de 10 pontos. A pontuação final será o somatório de pontos do jogador em cada diálogo. A pontuação é por empresa, ou seja, o jogador poderá ter uma pontuação para cada uma de suas empresas. A pontuação só será armazenada se o jogador conseguir responder todos os diálogos do jogo e se a pontuação final for superior a pontuação anterior do jogador para aquela empresa. Para montar o ranking do jogo será utilizada a pontuação de todas as empresas do jogo.

3 Enredo, Universo e Personagens

3.1 Enredo

A história do jogo se passa em uma cidade chamada SPI City, conhecida pela grande quantidade de empresas de desenvolvimento de software de altíssima qualidade instaladas, o que tem atraído muitos investidores para a cidade nos últimos anos. Neste contexto, o jogador receberá um investimento para montar sua própria empresa, mas como exigência, ele deverá alcançar o nível G de maturidade do MPS.BR no prazo máximo de um ano. A partir daí, a história se passa no interior de uma empresa de desenvolvimento de software fictícia, e o jogador no papel de gerente e dono da empresa, deverá resolver situações relacionadas com os resultados esperados do MPS.BR que serão apresentadas durante o jogo (ver Seção 4.3).

3.2 Personagens

- **Gerente da empresa:** papel do jogador no jogo, sendo responsável por gerenciar a sua empresa, resolvendo as situações apresentadas durante o jogo;
- **Secretária:** personagem do jogo responsável por informar o jogador sobre as tarefas que ele deve realizar no jogo;
- **Contador:** personagem do jogo responsável por informar o jogador sobre a situação financeira da empresa;
- **Desenvolvedores:** personagens do jogo responsáveis pelo desenvolvimento do produto da empresa;
- **Gerente de projetos:** personagem do jogo responsável por controlar os projetos da empresa e por informar o jogador sobre a situação dos projetos;
- **Implementador:** personagem do jogo responsável por fornecer *feedback* nos diálogos do jogo e por informar o jogador sobre os resultados esperados atendidos; e
- **Consultor Humphrey:** personagem do jogo responsável por esclarecer as dúvidas dos jogadores sobre conceitos de melhoria de processo de software.

4 Interface

4.1 Sistema Visual

O sistema visual do jogo será composto por gráficos e animações em 2D.

4.2 Sistema de Controle

A interação com o jogo será praticamente toda através do mouse, só será utilizado o teclado para interação com consultor Humphrey.

4.3 Sistema de Áudio

Inicialmente o jogo não possuirá sistema de áudio.

4.4 Sistema de Ajuda

No menu principal e no menu da tela principal do jogo terá a opção “tutorial”, que o jogador poderá acessar para obter informações sobre como jogar.

5 Projeto Técnico

5.1 Equipamento-alvo

O jogo é projetado para ser executado em qualquer computador com acesso a internet e com resolução mínima de 1024 por 768.

5.2 Ambiente de Desenvolvimento

A plataforma de desenvolvimento do jogo é composta pelo sistema operacional Windows 7, com Adobe Flash Professional CS5.5 para o desenvolvimento da interface e o FlashDevelop 4.0 para a programação do jogo. Para o desenvolvimento da parte do servidor será utilizado também o NetBeans 7.0 para programação em PHP e o phpMyAdmin 3.3.9 como gerenciador de banco de dados MySQL.

5.3 Linguagem de Programação

O jogo será desenvolvido utilizando a linguagem de programação ActionScript 3.0. A parte do servidor será desenvolvida utilizando a linguagem de programação PHP 5 e o banco de dados MySQL 5.

4.2.2 Technical Design Document (TDD)

O TDD corresponde a concepção técnica do jogo, sendo composto pelos requisitos funcionais, não-funcionais, regras de negócio, modelo de classe e modelo de entidade-relacionamento.

4.2.2.1 Requisitos Funcionais

Esta seção apresenta os requisitos funcionais do jogo elaborados a partir das informações do GDD. Os requisitos funcionais do jogo são apresentados a seguir:

- RF1. O jogo deve permitir que uma pessoa se cadastre como jogador;
- RF2. O jogo deve permitir que um jogador cadastrado faça login para ter acesso ao mesmo;
- RF3. O jogo deve permitir ao jogador cadastrar empresas;
- RF4. O jogador deve permitir ao jogador visualizar informações de suas empresas;
- RF5. O jogo deve permitir ao jogador selecionar uma empresa para começar a jogar;
- RF6. O jogo deve permitir ao jogador excluir suas empresas;
- RF7. O jogo deve permitir ao jogador visualizar as regras do mesmo;
- RF8. O jogo deve permitir ao jogador visualizar informações sobre a sua equipe de desenvolvimento;
- RF9. O jogo deve possuir um tutorial para que permita o jogador obter informações de como jogar;
- RF10. O jogo deve permitir ao jogador sair do mesmo a qualquer momento;
- RF11. O jogo deve permitir um jogador parar, continuar e avançar o progresso do jogo;
- RF12. O jogo deve permitir ao jogador consultar informações sobre os gastos e receitas de cada mês;
- RF13. O jogo deve permitir ao jogador consultar as tarefas que ele deve executar;
- RF14. O jogo deve permitir ao jogador consultar os resultados esperados atendidos;
- RF15. O jogo deve permitir ao jogador contratar desenvolvedores;
- RF16. O jogo deve permitir ao jogador demitir desenvolvedores;
- RF17. O jogo deve permitir ao jogador treinar os desenvolvedores;
- RF18. O jogo deve permitir ao jogador recompensar os desenvolvedores;
- RF19. O jogo deve permitir ao jogador consultar informações sobre o andamento dos projetos de sua empresa;
- RF20. O jogo deve permitir ao jogador realizar reuniões de acompanhamento do projeto;

- RF21. O jogo deve permitir ao jogador conversar com os desenvolvedores ou com o gerente de projetos para resolver uma situação;
- RF22. O jogo deve permitir ao jogador esclarecer suas dúvidas sobre conceitos de melhoria de processo de software;
- RF23. O jogo deve permitir ao jogador consultar o ranking de empresas.

4.2.2.2 Requisitos Não-Funcionais

Esta seção apresenta os requisitos não-funcionais do jogo elaborados a partir das informações do GDD. Os requisitos não-funcionais do jogo são apresentados a seguir:

- RNF1. O jogo deve ser executado nos navegadores Mozilla Firefox (versão 4.0), Internet Explorer (versão 8.0) e Google Chrome (versão 17);
- RNF2. O jogo deve ser desenvolvido utilizando a linguagem de programação ActionScript versão 3.0 e PHP versão 5;
- RNF3. O jogo deve utilizar o banco de dados MySQL versão 5;
- RNF4. O jogo deve poder ser executado em um computador com no mínimo um processador Pentium IV 1 GHz ou Athlon equivalente e com 256 MB de RAM;
- RNF5. O jogo deve ser executado em um computador com resolução mínima de 1024 por 768; e
- RNF7. O jogo não possuirá efeitos sonoros.

4.2.2.3 Regras do Jogo

Esta seção apresenta as regras do jogo elaboradas a partir das informações do GDD. As regras do jogo são apresentadas a seguir:

- O jogador poderá cadastrar até cinco empresas diferentes (ver Item 2.1.1 do GDD);
- O jogador poderá consultar suas tarefas sempre que desejar (ver Item 2.2.2.1 do GDD);
- O jogador poderá realizar somente uma tarefa por vez (ver Item 2.2.7 do GDD);

- O jogador poderá selecionar somente uma opção como resposta para um diálogo (ver Item 2.2.7 do GDD);
- O jogador deve aguardar um dia para poder continuar um diálogo que foi finalizado com erro (ver Item 2.2.7 do GDD);
- O jogador não poderá refazer uma tarefa após sua conclusão (ver Item 2.2.7 do GDD);
- O jogo será finalizado com fracasso caso o dinheiro da empresa fique negativo, ou caso o jogador não consiga completar todas as tarefas disponíveis no prazo de um ano (ver Item 2.1.1 do GDD);
- O jogo será completado com sucesso caso o jogador consiga realizar todas as tarefas disponíveis antes que se passe um ano (no tempo do jogo) e o dinheiro de sua empresa não fique negativo nesse período (ver Item 2.1.1 do GDD);
- O jogador deve contratar pelo menos um analista e um programador para poder iniciar um projeto (ver Item 2.2.1.1 do GDD);
- O jogador somente poderá demitir e contratar desenvolvedores (ver Item 2.2.1.1 do GDD);
- O jogador poderá contratar até quatro desenvolvedores para sua equipe de desenvolvimento (ver Item 2.2.1.1 do GDD);
- O jogador poderá demitir desenvolvedores sempre que desejar, porém o custo para demissão de um desenvolvedor será R\$ 3000,00 (ver Item 2.2.1.1 do GDD);
- O jogador poderá treinar seus desenvolvedores sempre que desejar, não havendo nenhum custo financeiro para empresa, mas o desenvolvedor ficará dois dias sem produzir (ver Item 2.2.1.2 do GDD);
- O jogador poderá recompensar um desenvolvedor sempre que desejar, porém será descontado R\$ 500,00 do dinheiro da empresa a cada recompensa (ver Item 2.2.1.3 do GDD);
- O jogador pode realizar reuniões de acompanhamento sempre que desejar, não havendo nenhum custo financeiro para a empresa (ver Item 2.2.1.6 do GDD);

- O jogador poderá consultar a situação do projeto sempre que desejar, não havendo nenhum custo financeiro para a empresa (ver Item 2.2.1.5 do GDD);
- O jogador poderá consultar os gastos e receitas da empresa sempre que desejar, não havendo nenhum custo financeiro para a empresa (ver Item 2.2.2.2 do GDD);
- O jogador poderá consultar o relatório de implementação sempre que desejar, não havendo nenhum custo financeiro para a empresa (ver Item 2.2.2.3 do GDD);
- O jogador poderá parar e continuar o progresso do jogo sempre que desejar (ver Item 2.2.4 do GDD); e
- O jogador poderá conversar com o consultor Humphrey sempre que desejar (ver Item 2.2.3 do GDD).

4.2.2.4 Modelo de Classes

A Figura 9 apresenta o modelo de classes do jogo, elaborado a partir das informações do GDD.

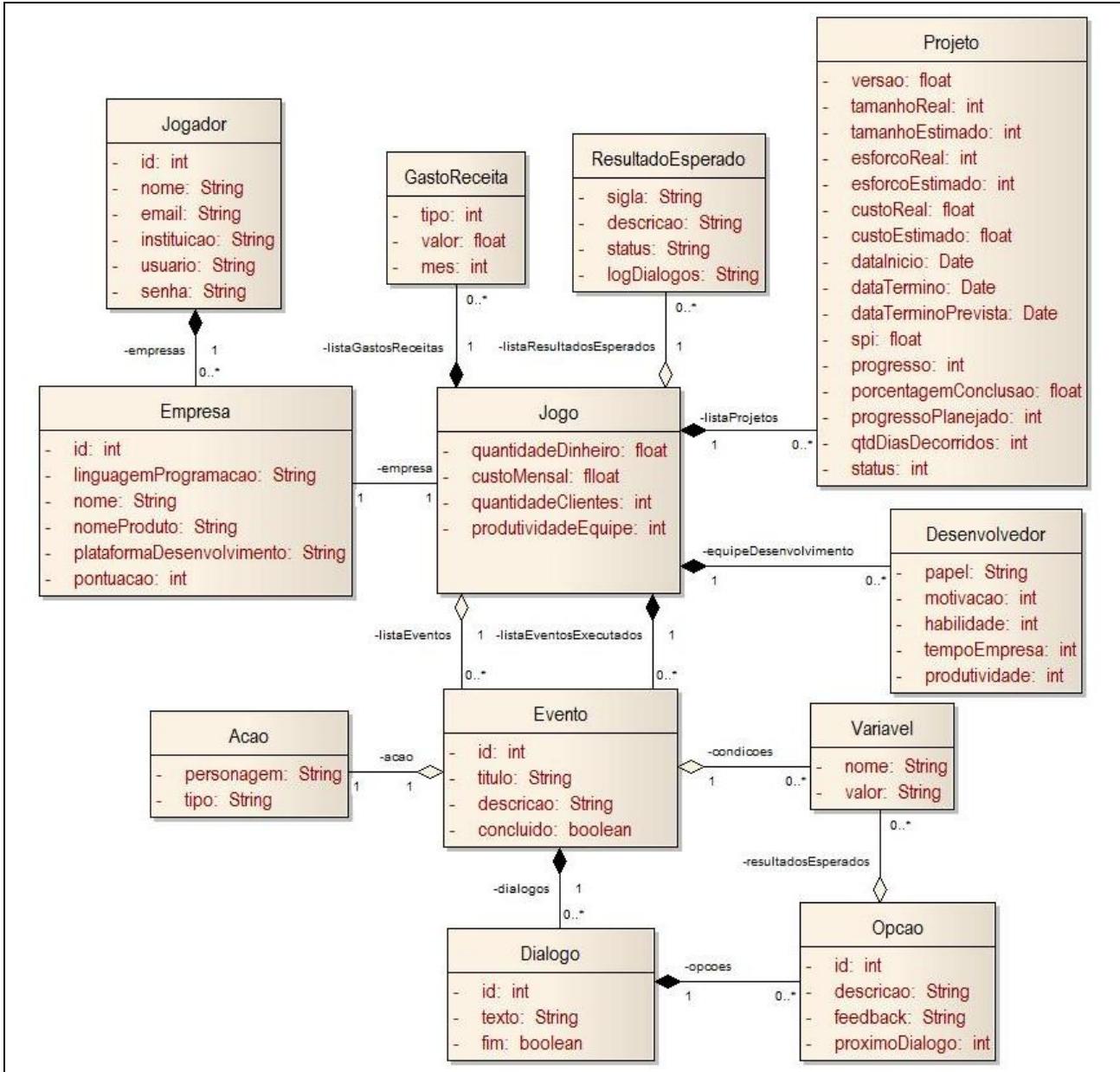


Figura 9. Modelo de classes

4.2.2.5 Modelo Entidade-Relacionamento

A Figura 10 apresenta o modelo de entidade-relacionamento do banco de dados do jogo, no qual é composto somente por duas entidades, pois nessa versão do jogo não serão armazenadas informações do progresso do jogador no jogo. A entidade “jogador” armazena informações dos jogadores cadastrados no jogo, e seu dicionário de dados é apresentado na Tabela 13. A entidade “empresa” armazena informações sobre as empresas cadastradas pelo jogador, e seu dicionário de dados é apresentado na Tabela 14.

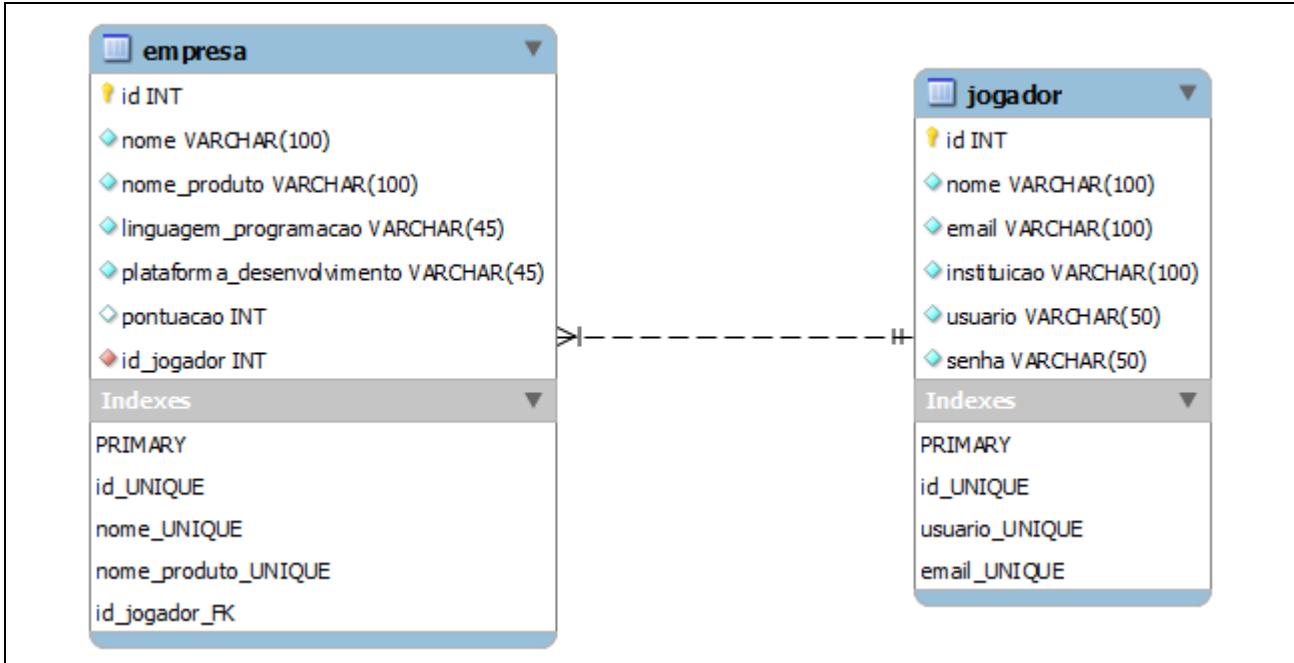


Figura 10. Modelo Entidade-Relacionamento

Tabela 13. Dicionário de dados da entidade empresa

Chave	Atributo	Tipo	Descrição
PK	id	int	Identificador único de uma empresa
	nome	varchar	Nome único de uma empresa
	nome_produto	varchar	Nome único do produto da empresa
	linguagem_programacao	varchar	Linguagem de programação utilizada para desenvolver o produto da empresa
	plataforma_desenvolvimento	varchar	Plataforma na qual o produto da empresa é desenvolvido, podendo ser: web, desktop ou móbil
	pontuacao	int	Pontuação obtida no jogo.
FK	id_jogador	int	Identificador do jogador que a empresa pertence

Tabela 14. Dicionário de dados da entidade jogador

Chave	Atributo	Tipo	Descrição
PK	id	int	Identificador único de um jogador
	nome	varchar	Nome do jogador
	email	varchar	Email do jogador
	instituicao	varchar	Instituição de ensino ou empresa que o jogador trabalha ou estuda
	usuario	varchar	Nome de acesso do jogador ao jogo
	senha	varchar	Senha de acesso do jogador ao jogo

4.3 EVENTOS DO JOGO

Os eventos do jogo são situações, relacionadas aos resultados esperados do MPS.BR, que o jogador terá que resolver durante o jogo. Esta seção apresenta como os eventos do jogo foram elaborados, os diálogos relacionados com cada resultado esperado do nível G de maturidade do MPS.BR, o nível de cobertura de cada resultado esperado pelos diálogos disponíveis no jogo, e a justificativa para o nível de cobertura.

Para elaboração dos diálogos primeiramente buscou-se por trabalhos que pudessem ajudar nessa tarefa. Para isso, foram realizadas pesquisas nos anais do SBQS (Simpósio Brasileiro de Qualidade de Software) e no Google, buscando por trabalhos que fornecessem informações de como os resultados esperados do nível G de maturidade do MPS.BR tinham sido implementados em empresas que implantaram o modelo MPS.BR. Nesta pesquisa foi encontrado somente um trabalho que trazia informações mais detalhadas de como os resultados esperados foram implementados (BERGMANN, 2008). A maioria dos trabalhos encontrados traziam informações de lições aprendidas, dificuldades e fatores de sucesso relacionados ao processo de implantação do MPS.BR de maneira geral, e não apresentavam informações específicas de como cada resultado foi implementado.

Como a pesquisa na literatura retornou poucos trabalhos que pudessem ajudar na elaboração dos diálogos, esta tarefa foi realizada na maior parte do tempo utilizando como base as informações contidas no guia de implementação do nível G de maturidade do MPS.BR (SOFTEX, 2011c) e o conhecimento de especialistas no modelo MPS.BR. Foram envolvidos na elaboração dos diálogos dois especialistas que atuam a mais de cinco anos com implementações do modelo MPS.BR, sendo que um deles é o orientador desta pesquisa.

O objetivo dos eventos do jogo é trazer situações do dia a dia de empresas de desenvolvimento de software, em que os resultados esperados do MPS.BR fossem indiretamente discutidos. Assim, espera-se que os jogadores possam entender com mais facilidade o que é necessário realizar em cada resultado esperado, pois os mesmos estariam associados com situações do seu dia a dia.

Foram elaborados diálogos somente para os resultados esperados dos processos do nível G de maturidade do MPS.BR. Não foram elaborados diálogos para os resultados esperados de atributo de processo (RAP). Até o presente momento, o jogo conta com pouco mais de 60 diálogos, distribuídos entre 6 situações diferentes. Essas situações e os diálogos de cada situação são apresentados no Apêndice B.

Na primeira situação do jogo os desenvolvedores estão com dúvida sobre como eles devem trabalhar, e o jogador deve conversar com eles para mostrar o que eles devem fazer primeiro. Nesta situação é discutido que antes de iniciar um projeto deve-se definir o escopo do mesmo, é explicado o que é escopo e a diferença entre escopo de projeto e escopo de produto, são apresentados exemplos de escopo de projeto e escopo de produto, é discutido sobre a necessidade de definição de um ciclo de vida para o projeto e sobre a importância de ter um ciclo de vida definido.

Na segunda situação o gerente de projetos está com dúvida sobre qual ciclo de vida utilizar, e o jogador deve conversar com ele para ajudá-lo a definir o ciclo de vida que será utilizado em seus projetos. Nesta situação é discutido sobre qual ciclo de vida é mais adequado para as características dos projetos da empresa, é discutido sobre a necessidade de definição das fases e atividades do ciclo de vida, são apresentados exemplos de fases de ciclo de vida, é discutido sobre a necessidade de se definir um responsável para cada atividade do projeto e que essa definição deve ser feita baseada nas habilidades e competências necessárias para executar cada atividade, é explicado a diferença entre habilidade e competência, e são apresentados exemplos de habilidades e competências.

Na terceira situação o gerente de projeto está com dúvida sobre o que colocar no plano de projeto, e o jogador deve conversar com ele para ajudá-lo a definir o que deve ser colocado no plano de projeto. Nesta situação é discutido sobre a necessidade de estimar o tamanho do projeto, é explicado o que é tamanho, é apresentada uma forma de medir tamanho, é discutido sobre a necessidade de estimar o esforço das atividades, é apresentada uma forma de medir o esforço, é discutido sobre a necessidade de estimar o custo das atividades, mas que o custo não precisa ser em valores monetários, é discutido sobre a necessidade de elaborar o cronograma do projeto definindo também os marcos do projeto, é explicado o que são marcos e porque é importante definir marcos, são apresentados exemplos de marcos, e é discutido sobre a necessidade de elaborar o orçamento do projeto.

Na quarta situação um pedido de alteração de requisitos é repassado para os desenvolvedores, e o jogador deve conversar com eles para decidir como tratar esse pedido de alteração. Nesta situação é discutido sobre a necessidade de registrar as solicitações de mudanças, sobre a necessidade de avaliar e registrar o impacto de uma mudança e que a rastreabilidade pode ser utilizada para apoiar a avaliação de impacto de uma mudança, é discutido sobre a necessidade de verificar se a mudança irá afetar no custo e no cronograma do projeto, sobre a necessidade de aprovação da mudança junto ao fornecedor de requisitos, sobre a necessidade de revisar os produtos de trabalho do projeto após a aprovação de uma mudança para corrigir possíveis inconsistências em relação aos requisitos, e sobre a necessidade de revisar os requisitos e o plano com a equipe após uma mudança aprovada para manter o comprometimento da equipe.

Na quinta situação o gerente de projetos está com dúvida em como os riscos devem ser planejados, e o gerente de projetos deve conversar com ele para definir como os riscos serão planejados. Nesta situação é discutido sobre a necessidade de identificar e registrar os riscos do projeto, definindo a probabilidade de ocorrência e o impacto de cada risco, é discutido que os riscos precisam ser priorizados, que para o nível G não é necessário ter um plano de respostas aos riscos, que os riscos identificados precisam ser acompanhados, que os problemas causados pela ocorrência de um risco precisam ser analisados e registrados, e que ações para corrigir ou impedir que os problemas causados pelos riscos voltem a ocorrer precisam ser planejadas e acompanhadas até a sua conclusão.

Na sexta e última situação disponível no jogo até o momento, o gerente de projetos está com um problema em relação aos artefatos do projeto, e o jogador deve conversar com ele para ajudá-lo a resolver este problema. Nesta situação é discutido sobre a necessidade de planejar os dados do projeto, é explicado o que é dado e são apresentados exemplos de dados de projeto, é apresentada uma forma de planejar os dados do projeto, são apresentados exemplos de informações que podem ser colocadas no plano de dados, e é discutido que o plano de comunicação pode estar integrado com o plano de dados do projeto.

Para demonstrar os diálogos relacionados com cada resultado esperado, foi definido um identificador para cada diálogo, que segue o seguinte padrão: EV.<número da situação>.<número do diálogo>. Na Tabela 15 é apresentado todos os resultados esperados dos processos do nível G de maturidade do MPS.BR, os diálogos relacionados com cada resultado esperado, o nível de cobertura dos resultados esperados pelos diálogos do jogo, e a justificativa sobre o nível de cobertura.

Tabela 15. Diálogos relacionados com cada resultado esperado, grau de cobertura e justificativa

Resultado Esperado	Diálogos	Nível de cobertura	Justificativa
GPR1	EV.1.1 EV.1.2 EV.1.3 EV.1.4	Totalmente coberto	Este resultado está totalmente coberto porque é discutida a necessidade de se ter um escopo definido antes de iniciar um projeto (EV.1.1), é explicado o que é escopo e a diferença entre escopo de projeto e de produto (EV.1.2), e são apresentados exemplos de escopo de projeto (EV.1.4) e escopo de produto (EV.1.3).
GPR2	EV.3.2 EV.3.3 EV.3.4	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que no planejamento do projeto deve ser estimado o tamanho do projeto (EV.3.2), é explicado o que é tamanho e a importância da estimativa de tamanho (EV.3.2, EV.3.3), e é apresentada uma maneira de medir o tamanho do projeto (EV.3.3, EV.3.4).
GPR3	EV.1.5 EV.1.6 EV.2.1 EV.2.2 EV.2.3 EV.2.4 EV.2.5 EV.2.6 EV.2.7	Totalmente coberto	Este resultado está totalmente coberto porque é discutida a necessidade de se ter um ciclo de vida para o projeto (EV.1.5) e a importância de se ter um ciclo de vida definido (EV.1.6), é discutido que o ciclo de vida deve ser adequado às características do projeto (EV.2.1, EV.2.2, EV.2.3), são apresentados exemplos de ciclo de vida e para quais características eles são mais adequados (EV.2.2, EV.2.3, EV.2.4), é discutido que além de ter o ciclo de vida é importante definir as fases e as principais atividades do ciclo de vida (EV.2.5, EV.2.7), e são apresentados exemplos de fases de ciclo de vida de projeto (EV.2.6).
GPR4	EV.3.5 EV.3.6 EV.3.7 EV.3.8 EV.3.9	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que no planejamento do projeto deve ser estimado o esforço das atividades do projeto (EV.3.5), é apresentada uma maneira de estimar o esforço das atividades (EV.3.6), é discutido que além de estimar o esforço deve ser estimado também o custo das atividades (EV.3.8), é discutido que o custo não precisa ser em valores monetários e que neste caso o custo das atividades pode ser considerado igual ao seu esforço (EV.3.9).
GPR5	EV.3.10 EV.3.11 EV.3.12 EV.3.13 EV.3.14 EV.4.6	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que no planejamento do projeto deve ser elaborado o cronograma para o projeto incluindo a definição dos marcos (EV.3.10), é discutido o que são marcos e por que eles são importantes (EV.3.11, EV.3.12), são apresentados exemplos de marcos (EV.3.11), é discutido que o orçamento para o projeto também precisa ser elaborado (EV.3.13), mesmo que o gerente de projetos não gerencie os custos financeiros do projeto (EV.3.14), e é discutido que o cronograma e o orçamento precisam ser ajustados sempre que necessário (EV.4.6).
GPR6	EV.3.15 EV.5.1 EV.5.2 EV.5.3 EV.5.4 EV.5.5	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que os riscos do projeto precisam ser identificados (EV.3.15 EV.5.1), e que o seu impacto, probabilidade de ocorrência e prioridade precisam ser determinados e registrados (EV.5.2, EV.5.3), é discutido que para o nível G não é necessário ter um plano de respostas aos riscos (EV.5.4), e é discutido que o documento de riscos do projeto precisa ser acompanhado e atualizado durante a execução do projeto (EV.5.5).
GPR7	EV.1.7	Totalmente	Este resultado está totalmente coberto porque é discutido que os

	EV.2.8 EV.2.9 EV.2.10 EV.2.11 EV.2.12	coberto	responsáveis por executar cada atividade do projeto precisam ser determinados (EV.1.7, EV.2.8, EV.2.9), que para cada atividade deve ser determinado os conhecimentos e habilidades necessários para executá-la (EV.2.9), que para saber quem pode executar cada atividade deve haver um documento com as habilidades e competências de cada colaborador (EV.2.9), são apresentados exemplos de habilidades e competências (EV.2.10, EV.2.11), e é discutido que deve-se contratar ou planejar um treinamento quando não houver alguém com o perfil necessário para executar uma atividade (EV.2.12).
GPR8	EV.1.7 EV.3.15	Parcialmente coberto	Este resultado está parcialmente coberto porque só é discutido que os recursos físicos precisam ser planejados (EV.1.7, EV. 3.15), porém ainda não há nenhum diálogo que discuta o planejamento dos recursos físicos e ambiente.
GPR9	EV.1.7 EV.3.15 EV.6.2 EV.6.3 EV.6.4 EV.6.5 EV.6.6 EV.6.7	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que os dados do projeto precisam ser planejados (EV.1.7, EV.3.15 EV.6.2), é discutido o que são dados do projeto (EV.6.3), é discutido por que é importante planejar os dados do projeto (EV.6.4), é apresentado uma forma de planejar os dados do projeto (EV.6.5), e são apresentados exemplos de informações que devem constar no plano de dados do projeto, incluindo informações de privacidade e segurança (EV.6.6, EV.6.7).
GPR10	EV.3.7 EV.3.9 EV.3.11 EV.3.16 EV.4.8 EV.4.9 EV.5.5 EV.5.6 EV.6.3	Totalmente coberto	Este resultado está totalmente coberto porque em vários diálogos são feitas referências ao plano de projeto.
GPR11	EV.3.12	Parcialmente coberto	Este resultado está parcialmente coberto porque é discutido somente que nas revisões de marcos do projeto pode ser questionado a viabilidade de continuar o projeto (EV.3.12), porém ainda não há nenhum diálogo que discuta a análise de viabilidade no início do projeto e como essa análise de viabilidade pode ser realizada.
GPR12	EV.3.16 EV.4.9	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que o plano de projeto deve ser revisado com todos os interessados para obter o comprometimento deles com o plano (EV.3.16), e é discutido que sempre que houver uma mudança no plano de projeto, as mudanças precisam ser revisadas com todos os afetados para que o comprometimento com o plano seja mantido (EV.4.9).
GPR13	EV.3.16	Parcialmente coberto	Este resultado está parcialmente coberto porque é discutido que o plano de projeto precisa ser monitorado (EV.3.16), mas ainda não há um diálogo que discuta o monitoramento do projeto de maneira mais detalhada.
GPR14	EV.3.16 EV.6.4	Parcialmente atendido	Este resultado está parcialmente atendido porque é discutido somente que os dados precisam ser monitorados (EV.6.4), mas ainda não há nenhum diálogo que discuta o monitoramento de recursos matérias e humanos.

GPR15	EV.3.16 EV.5.5 EV.5.6 EV.5.7	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que o documento de riscos do projeto precisa ser monitorado e atualizado (EV.3.16, EV.5.5), que os problemas causados pela ocorrência de um risco precisam ser registrados, e que ações para corrigir os problemas identificados precisam ser planejadas e acompanhadas até que os problemas sejam resolvidos (EV.5.6, EV.5.7).
GPR16	EV.6.7	Parcialmente coberto	Este resultado está parcialmente coberto porque é discutido que o plano de comunicação pode ser integrado ao plano de dados, adicionando algumas informações que evidenciam também a comunicação dos envolvidos (EV.6.7), porém ainda estão faltando diálogos para discutir a comunicação mais detalhadamente.
GPR17	EV.3.10 EV.3.11 EV.3.12	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que o cronograma do projeto deve incluir a definição dos marcos (EV.3.10), são apresentados exemplos de marcos (EV.3.11), é discutido que revisões devem ser realizadas nos marcos do projeto e a importância de se realizar revisões em marcos, e são apresentados exemplos do que deve ser verificado nas revisões de marcos (EV.3.12).
GPR18	EV.5.6 EV.5.7	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que os problemas identificados durante o projeto precisam ser analisados, junto com as partes interessadas e registrados (EV.5.6), e que ações para corrigir estes problemas devem ser planejadas e acompanhadas até que os problemas sejam resolvidos (EV.5.6, EV.5.7).
GPR19	EV.5.6 EV.5.7	Totalmente coberto	Este resultado está totalmente coberto porque é discutido que ações para corrigir os problemas identificados durante o projeto ou para prevenir a repetição desses problemas devem ser planejadas e acompanhadas até que elas sejam resolvidas (EV.5.6, EV.5.7).
GRE1	EV.4.3 EV.4.7	Parcialmente coberto	Este resultado está parcialmente coberto porque em alguns diálogos é mencionado a existência de um documento de requisitos aprovado (EV.4.3, EV.4.7), porém ainda faltam diálogos para discutir o entendimento dos requisitos de forma mais detalhada.
GRE2	EV.4.9	Parcialmente coberto	Este resultado está parcialmente coberto porque é discutido que os requisitos precisam ser revisados com a equipe técnica para manter o comprometimento dos mesmos depois que uma mudança de requisitos for aprovada (EV.4.9), porém ainda faltam diálogos que discutam de maneira mais detalhada a aceitação dos requisitos pela equipe técnica no início do projeto.
GRE3	EV.4.5 EV.4.8	Totalmente coberto	Este resultado está totalmente coberto porque é discutido sobre a utilização da rastreabilidade para apoiar a análise de impacto de uma mudança (EV.4.5), e é discutido sobre a necessidade de manter a rastreabilidade atualizada (EV.4.8).
GRE4	EV.4.4	Largamente coberto	Este resultado está largamente coberto porque é discutido que é necessário revisar os artefatos do projeto após a aprovação de uma mudança de requisitos para resolver possíveis inconsistências nos artefatos em relação aos requisitos (EV.4.4), porém ainda falta discutir que esta revisão precisa ser realizada também durante a monitoração do projeto.
GRE5	EV.4.2	Totalmente	Este resultado está totalmente coberto porque é discutido que as

	EV.4.3 EV.4.4 EV.4.5 EV.4.6 EV.4.7 EV.4.8 EV.4.9	coberto	solicitações de mudanças de requisitos precisam ser registradas (EV.4.2) e a importância delas estarem registradas (EV.4.3), é discutido que é necessário analisar e registrar o impacto das mudanças (EV.4.4) e que a rastreabilidade pode ser utilizada para apoiar a análise deste impacto (EV.4.5), que é necessário verificar se a mudança irá impactar no custo e no cronograma do projeto e que é necessário aprovar a mudança junto ao fornecedor de requisitos (EV.4.6), que é interessante reservar um tempo no cronograma do projeto para análise de solicitações de mudanças (EV.4.7), que é necessário revisar os artefatos afetados pela mudança após sua aprovação (EV.4.8), e que é necessário revisar o plano com a equipe após a aprovação da mudança se o mesmo for alterado (EV.4.9).
--	--	---------	---

Como pode ser observado na Tabela 15, todos os resultados esperados possuem pelo menos um diálogo associado, porém alguns resultados esperados ainda não estão completamente cobertos pelos eventos disponíveis no jogo até o momento. Porém, até o final dessa pesquisa serão elaborados mais diálogos, pelo menos para cobrir completamente todos os resultados esperados dos processos do nível de G de maturidade do MPS.BR.

4.4 SPI CITY

Esta seção apresenta o jogo para apoio ao ensino de melhoria de processo de software, denominado SPI City, através de uma breve descrição das telas do mesmo. O jogo pode ser acessado através do seguinte endereço <http://www.incremental.com.br/spicity>.

Ao acessar o jogo será exibida uma animação que apresenta a cidade de SPI City e conta que o jogador recebeu um investimento para montar sua empresa, mas como exigência ele deverá alcançar o nível G de maturidade do MPS.BR no prazo máximo de um ano. A Tabela 16 apresenta os textos de cada quadro desta animação e a Figura 11 apresenta as imagens desta animação.

Tabela 16. Sequência de quadros da animação inicial do jogo

Quadro	Diálogo
1	SPI City é uma cidade muito agitada conhecida pela grande quantidade de empresas de desenvolvimento de software de alta qualidade.
2	O que tem atraído muitos investidores para a cidade nos últimos anos, motivando o surgimento de muitas novas empresas.
3	Este crescimento tem aumentado também a competitividade na cidade, fazendo com que as empresas cada vez mais se preocupem com a melhoria de seus processos.
4	Você acaba de receber um investimento para montar sua própria empresa, mas com uma restrição você deve alcançar o nível G do MPS.BR no prazo máximo de um ano.
5	Agora monte sua empresa, mostre que você é bom em melhoria de processo de software e desponha no ranking das melhores empresas de SPI City.



Figura 11. Animação inicial do jogo

Após a animação será exibida a tela de login (Figura 12-A) na qual o jogador deverá informar seu usuário e sua senha para acessar o jogo, ou clicar em “cadastrar” caso ele ainda não possua cadastro. Selezionando a opção cadastro na tela de login, será exibida a tela de cadastro de jogador (Figura 12-B). Para se cadastrar no jogo devem ser informados os seguintes dados: nome, email, empresa/instituição, nome de usuário e senha. Após o cadastro do jogador ou caso o cadastro seja cancelado, o jogo retorna para tela de login.



Figura 12. Tela de login e tela de cadastro de jogador

Após a tela de login, caso o jogador tenha informado corretamente seus dados, será exibido o menu principal do jogo (Figura 13), com as seguintes opções: empresas, tutorial, sobre, regras, ranking de empresas, e sair. Selecionando a opção “sobre” será exibida uma tela com informações sobre o jogo e sobre os desenvolvedores do jogo (Figura 14-A). Selecionando a opção “regras” será exibida a tela com as regras do jogo (Figura 14-B). Selecionado a opção “tutorial” será exibido o tutorial do jogo (Figura 15). Selecionando a opção “ranking de empresas” será exibida a tela com o ranking de empresas do jogo (). Selecionando a opção “sair” o jogador perde acesso ao jogo e é redirecionado para a tela de login.



Figura 13. Menu principal do jogo

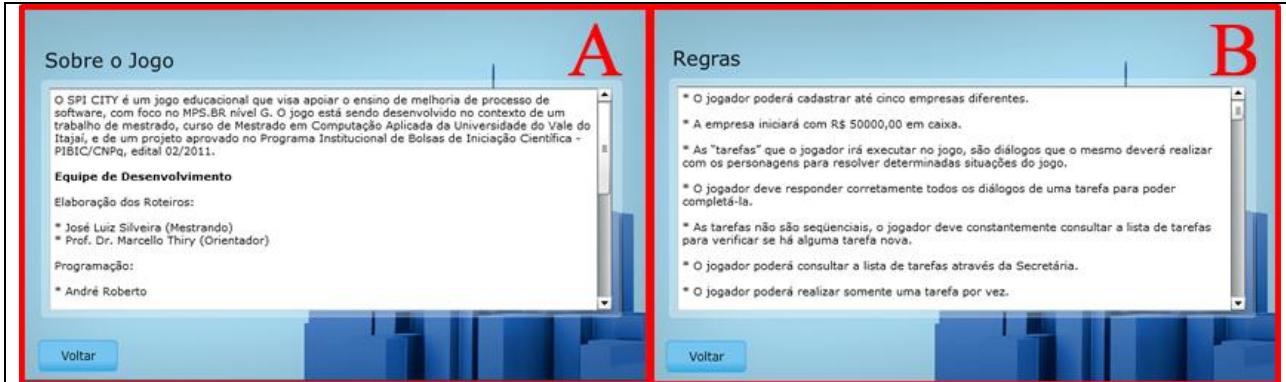


Figura 14. Tela de regras do jogo

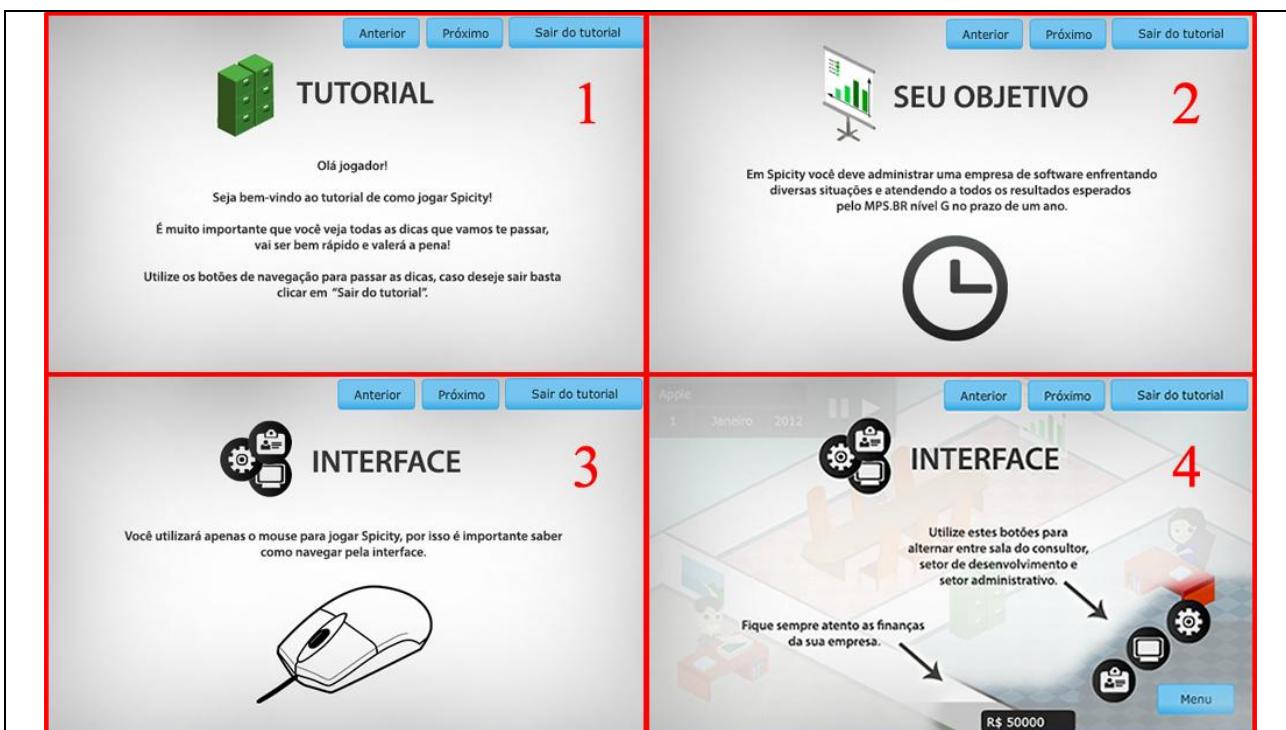


Figura 15. Algumas das telas do tutorial do jogo

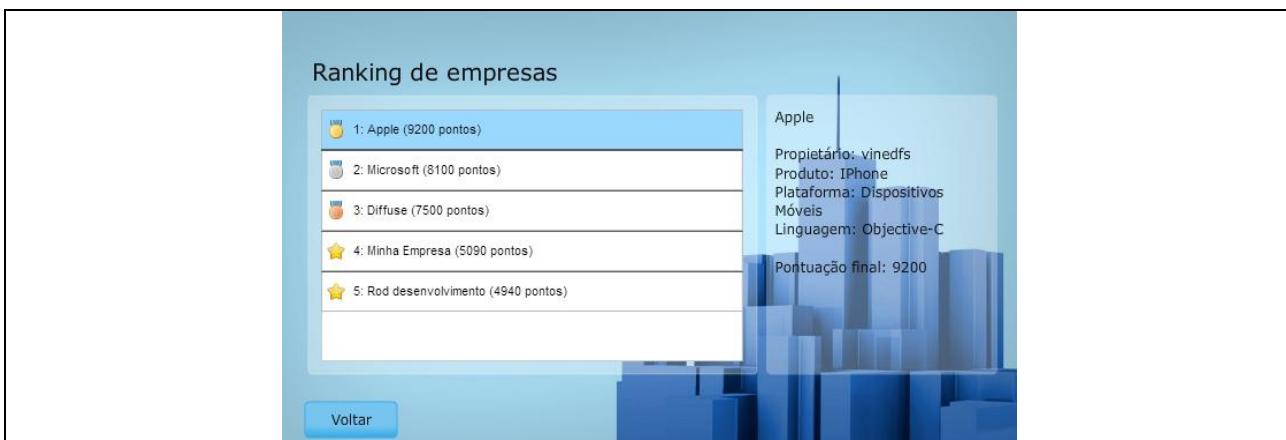


Figura 16. Tela de ranking de empresas do jogo

Selecionando a opção “empresas” no menu principal do jogo, será exibida a tela de listagem de empresas do jogador (Figura 17-A). Nesta tela o jogador pode visualizar as informações de suas empresas (clicando sobre uma empresa), pode cadastrar novas empresas, pode excluir uma empresa ou selecionar uma empresa para iniciar o jogo. Selecionando a opção “cadastrar” na tela de listagem de empresas, será exibida a tela de cadastro de empresas (Figura 17-B). Para cadastrar uma empresa devem ser informados os seguintes dados: nome da empresa, nome do produto, linguagem de programação utilizada e plataforma de desenvolvimento (web ou desktop).



Figura 17. Tela de listagem das empresas do jogador

Ao selecionar uma empresa para iniciar um jogo, na tela de listagem de empresas, será exibida a tela principal do jogo (Figura 18). A tela principal do jogo simula o ambiente interno de uma empresa de desenvolvimento de software, e é dividida em três áreas: setor administrativo (Figura 18-A), setor de desenvolvimento (Figura 18-B) e sala do consultor (Figura 18-C). Para navegar entre as áreas da empresa, o jogador deve utilizar os botões localizados na parte inferior direita da tela principal.

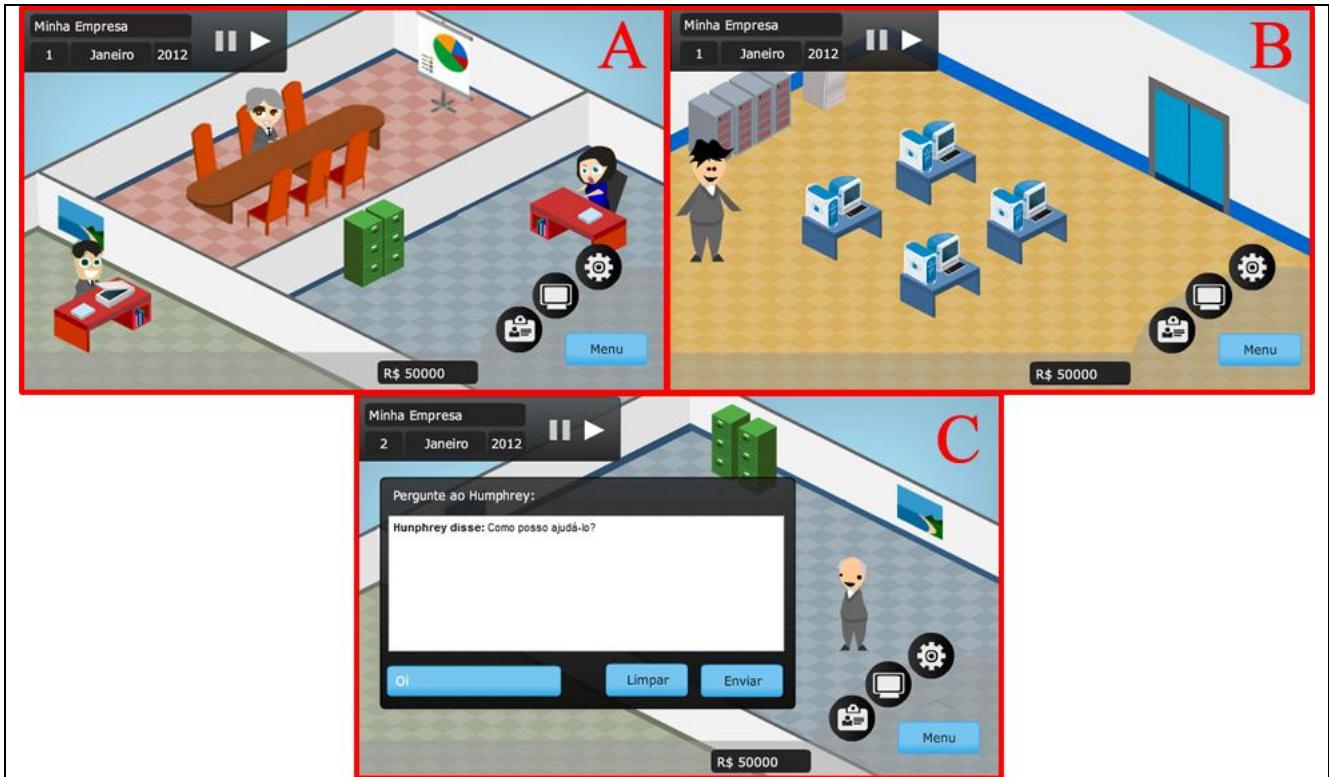


Figura 18. Tela principal do jogo

Após iniciar o jogo a primeira coisa que o jogador deve fazer é contratar desenvolvedores para compor sua equipe de desenvolvimento. Para que o produto da empresa comece a ser desenvolvido o jogador deve contratar pelo menos um analista e um programador. Para contratar, o jogador deve clicar sobre uma mesa vazia, depois selecionar a opção “contratar” (Figura 19-1), e depois selecionar o tipo de desenvolvedor que ele deseja contratar (Figura 19-2). Feito isso, um novo desenvolvedor é contratado e passa a utilizar a mesa que o jogador selecionou (Figura 19-3).

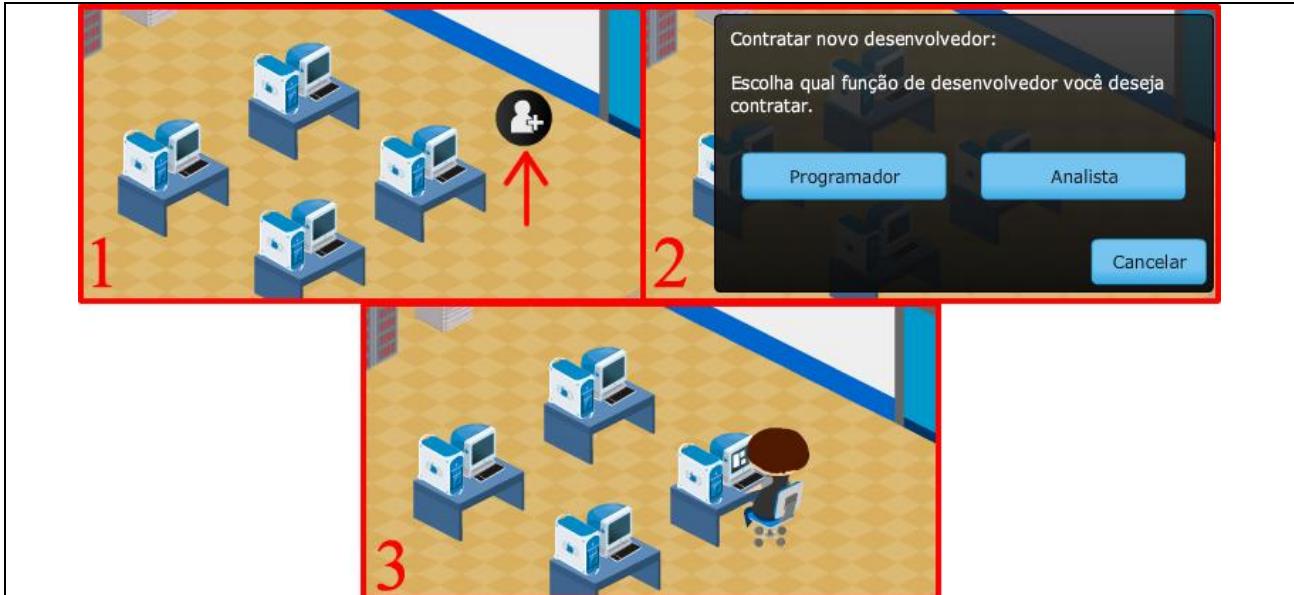


Figura 19. Contratar desenvolvedor

Depois de contratar um desenvolvedor o jogador pode treinar, recompensar ou demitir este desenvolvedor. Para treinar, o jogador deve clicar sobre o desenvolvedor (Figura 20-1), selecionar a opção “treinar” (Figura 20-2), e selecionar o tipo de treinamento que será realizado (Figura 20-3). Feito isso, o desenvolvedor selecionado ficará em treinamento por dois dias (Figura 20-4).

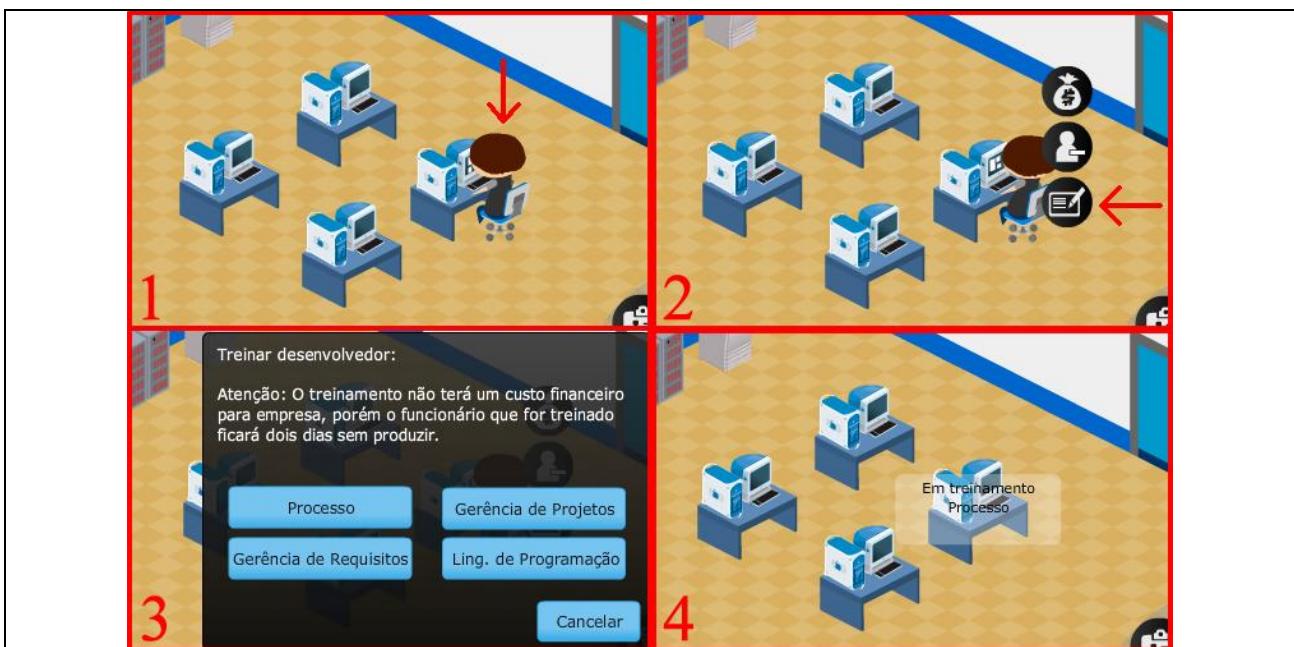


Figura 20. Treinar desenvolvedor

Para recompensar um desenvolvedor, o jogador deve clicar sobre o mesmo (Figura 21-1), selecionar a opção “recompensar” (Figura 21-2), e selecionar a opção “sim” na tela de confirmação de recompensa (Figura 21-3). Ao recompensar um desenvolvedor será descontado R\$ 500,00 do dinheiro da empresa (Figura 21-4).

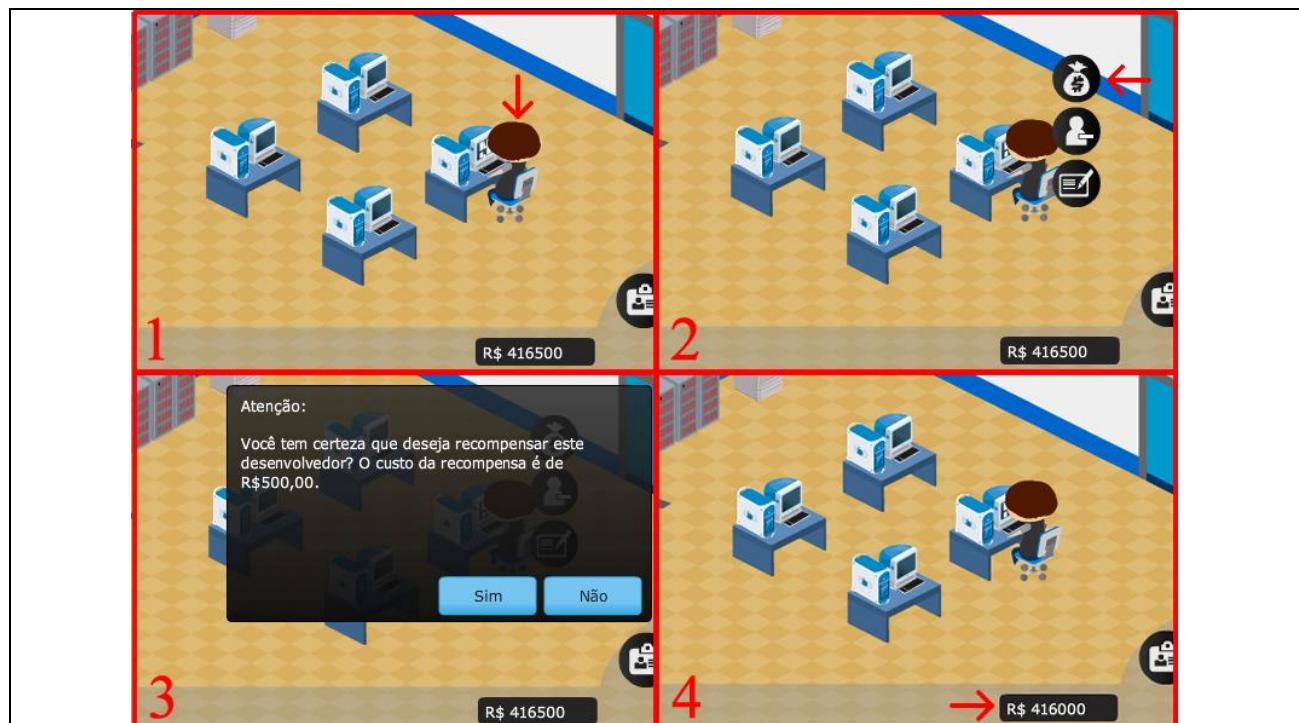


Figura 21. Recompensar desenvolvedor

Para demitir, o jogador deve clicar sobre um desenvolvedor (Figura 22-1), selecionar a opção ”demitir” (Figura 22-2), e selecionar a opção “sim” na tela de confirmação de demissão (Figura 22-3). Ao demitir um desenvolvedor será descontado R\$ 3000,00 do dinheiro da empresa e a mesa que o mesmo ocupava ficará vazia (Figura 22-4).



Figura 22. Demitir desenvolvedor

O jogador poderá acompanhar a situação dos projetos de sua empresa ou realizar reuniões de acompanhamento de projeto, através do gerente de projetos. Para visualizar a situação do projeto em desenvolvimento, o jogador deve clicar sobre o gerente de projetos (Figura 23-1) e selecionar a opção “status do projeto” (Figura 23-2). Feito isso, será exibida uma tela contendo as seguintes informações sobre o projeto em desenvolvimento (Figura 23-3): versão, estado (atrasado, adiantado, em dia), custo estimado, tamanho estimado, porcentagem de conclusão, etc.

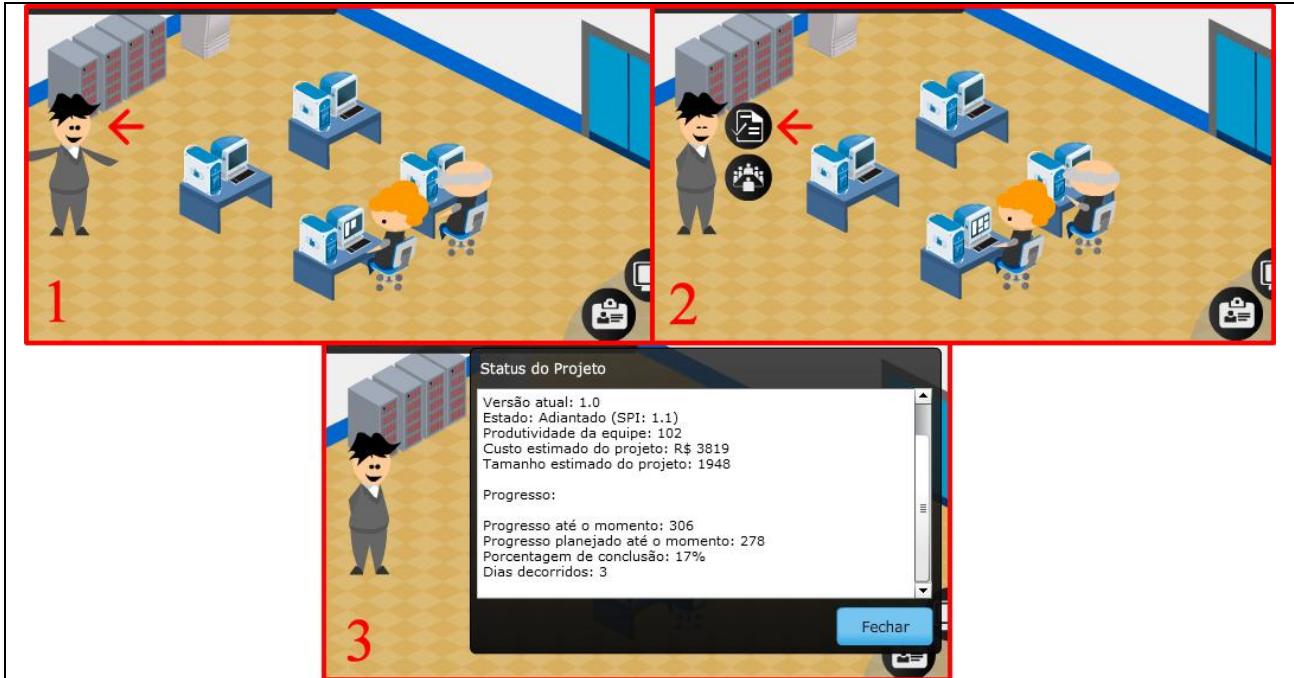


Figura 23. Consultar situação do projeto

As reuniões de acompanhamento de projeto são importantes para manter os desenvolvedores motivados. Para realizar uma reunião de acompanhamento de projeto o jogador deve clicar sobre o gerente de projeto (Figura 24-1) e selecionar a opção “reunião de acompanhamento” (Figura 24-2). Feito isso, será exibida uma tela avisando que a reunião de acompanhamento foi realizada (Figura 24-3).

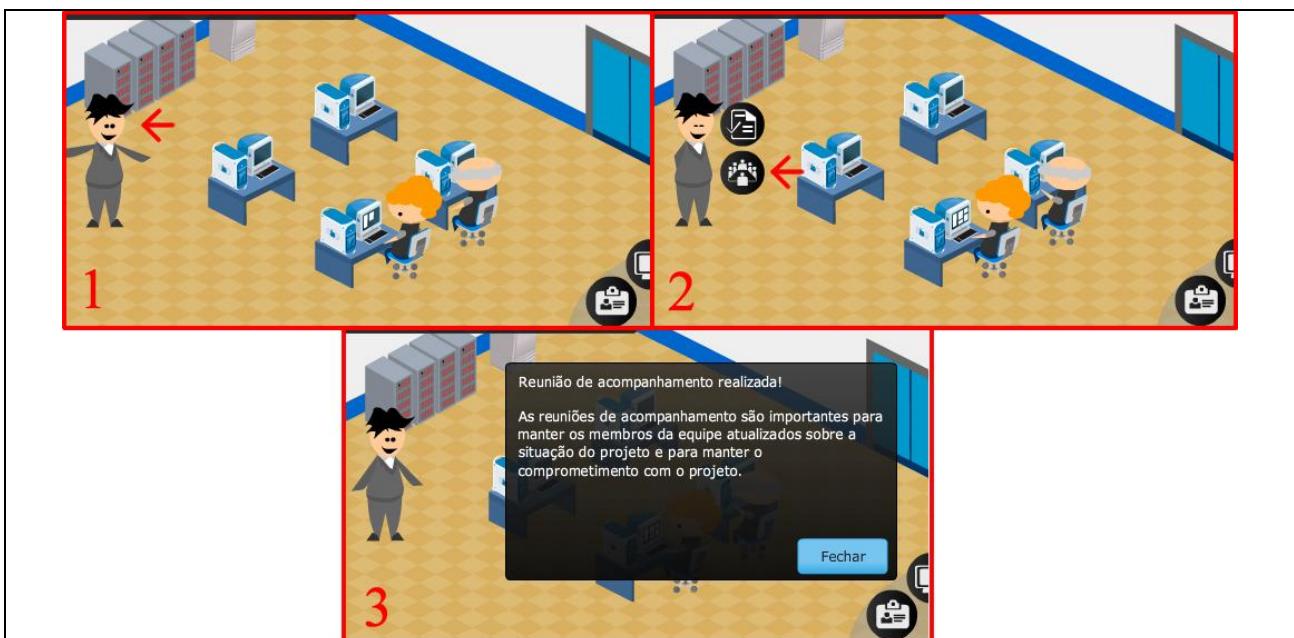


Figura 24. Reunião de acompanhamento de projeto

As tarefas do jogador são situações que ele terá que resolver durante o jogo. O jogador pode consultar suas tarefas através da secretaria, que fica no setor administrativo. Para consultar suas tarefas o jogador deve clicar sobre a secretaria e selecionar a opção “ver tarefas”. Feito isso, será exibida uma tela com a lista de tarefas que o jogador já resolveu, caso ele já tenha resolvido alguma, e com a tarefa que está pendente para o jogador resolver, caso tenha alguma (Figura 25). Clicando sobre uma tarefa da lista, o jogador pode visualizar o que deve ser realizado naquela tarefa.

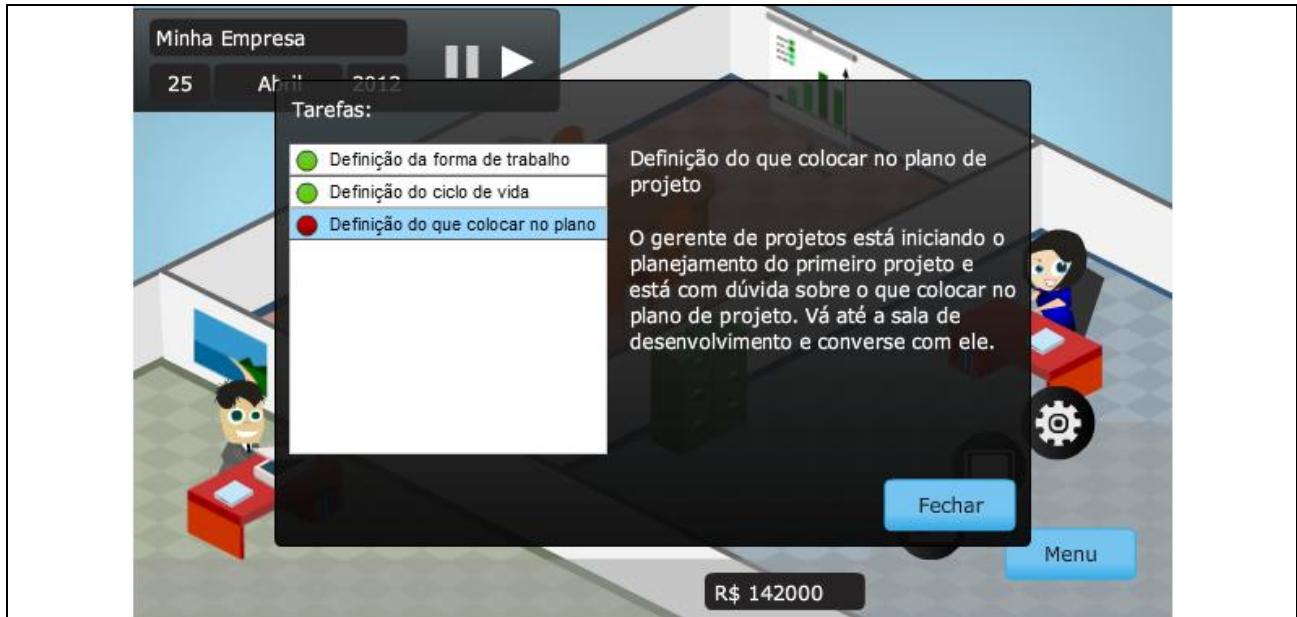


Figura 25. Visualizar tarefas

Para resolver uma tarefa o jogador deve conversar com o personagem associado com a mesma. Por exemplo, na tarefa apresentada na Figura 25 o jogador deve conversar com o gerente de projetos para ajudá-lo a definir o que colocar no plano de projeto. Para conversar com o personagem associado com a tarefa, o jogador deve clicar sobre o mesmo e selecionar a opção “conversar”. Feito isso, será exibida a tela de diálogo, simulando uma conversa entre o jogador e o personagem (Figura 26). Nesta tela o jogador deve selecionar uma opção como resposta para o diálogo. Após selecionar uma opção será exibida a tela de *feedback* (Figura 27), informando se a opção selecionada está correta e trazendo explicações adicionais sobre o diálogo. Se o jogador selecionou a opção incorreta o diálogo será encerrado após a tela de *feedback*, e o jogador terá que voltar a conversar com o personagem, após passar um dia no tempo do jogo, para continuar a resolver a tarefa. Se o jogador selecionou a opção correta, após a tela de *feedback* será exibido o próximo diálogo, até que se chegue no último diálogo da situação. Respondendo corretamente todos

os diálogos de uma situação a tarefa será concluída. Ao final de uma tarefa será exibida uma tela informando os resultados esperados atendidos naquela tarefa (Figura 28).

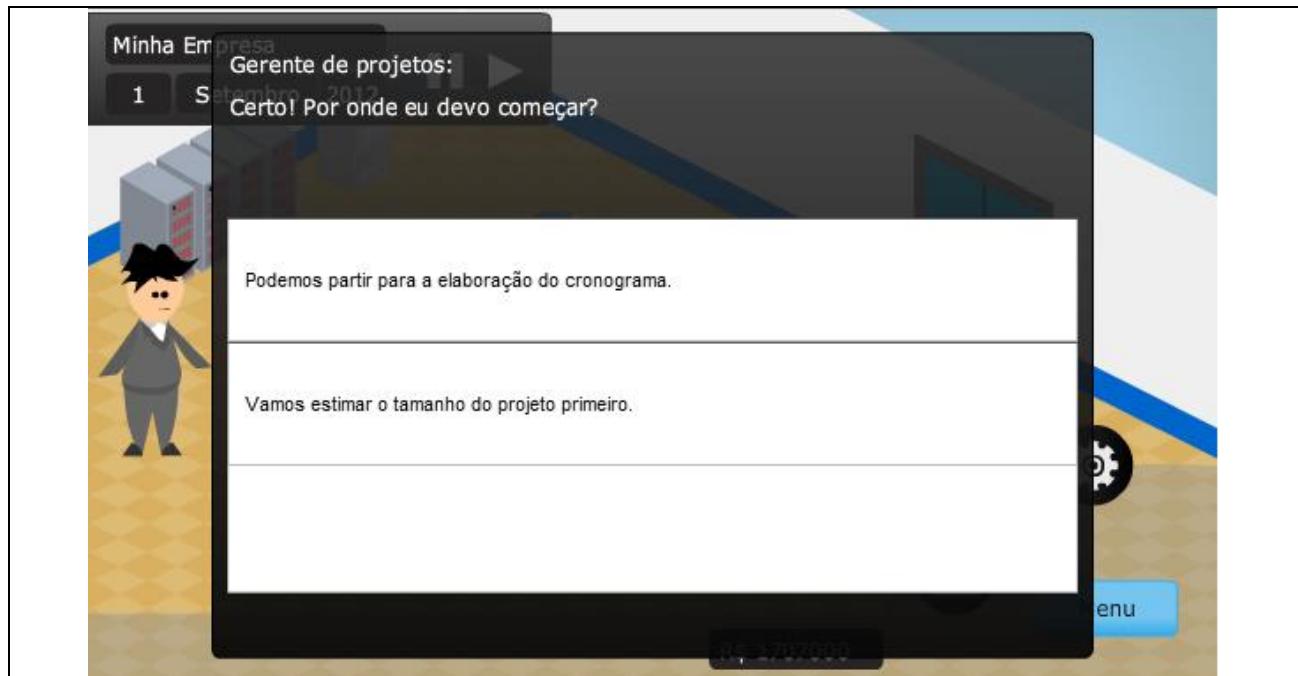


Figura 26. Tela de diálogo

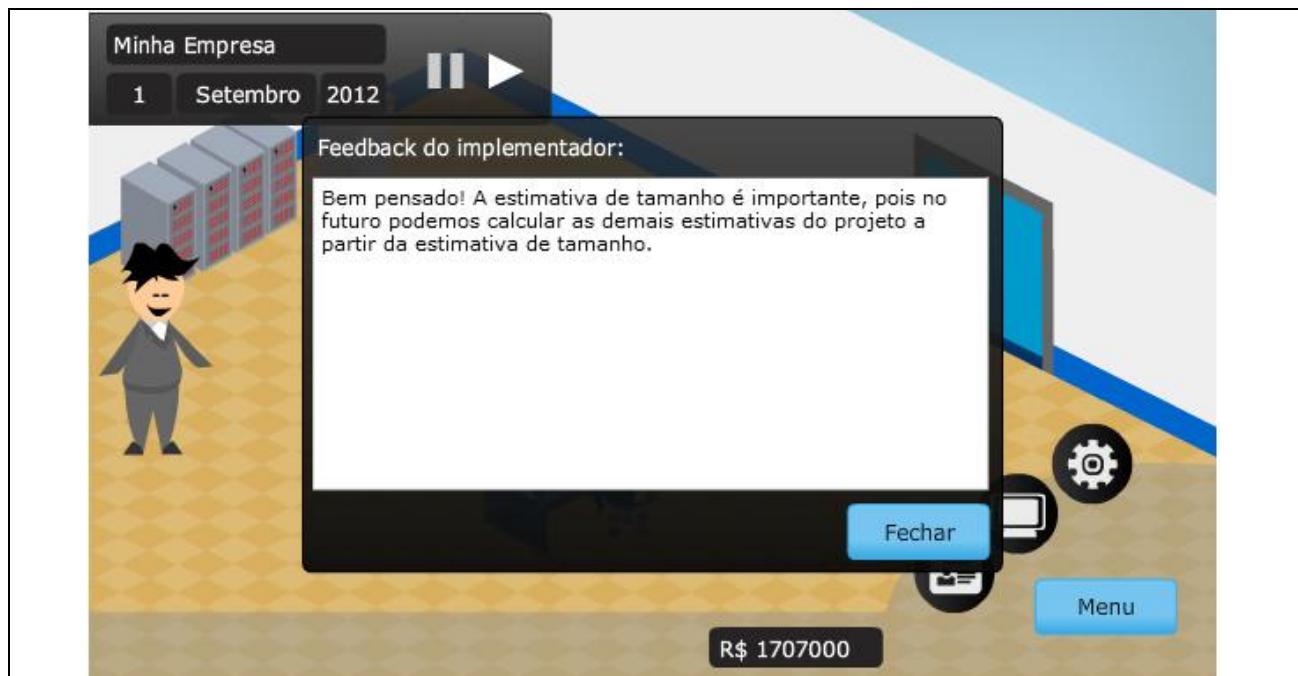


Figura 27. Tela de feedback

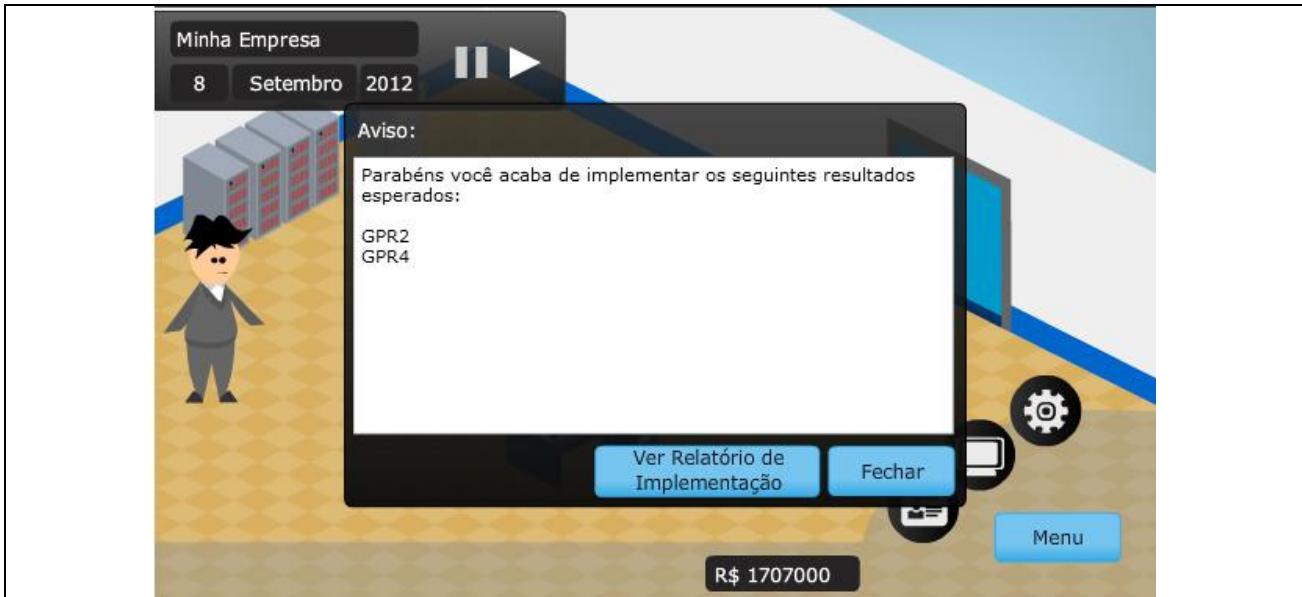


Figura 28. Tela de aviso de resultados esperados atendidos

Na tela de aviso de resultados esperados atendidos (Figura 28), o jogador pode selecionar a opção “ver relatório de implementação” para visualizar todos os resultados esperados atendidos até o momento, e os diálogos associados com cada resultado esperado atendido. O jogador também pode consultar o relatório de implementação através do implementador, que fica no setor administrativo, clicando sobre o mesmo e selecionando a opção “ver relatório de implementação”. A Figura 29 apresenta a tela de relatório de implementação.

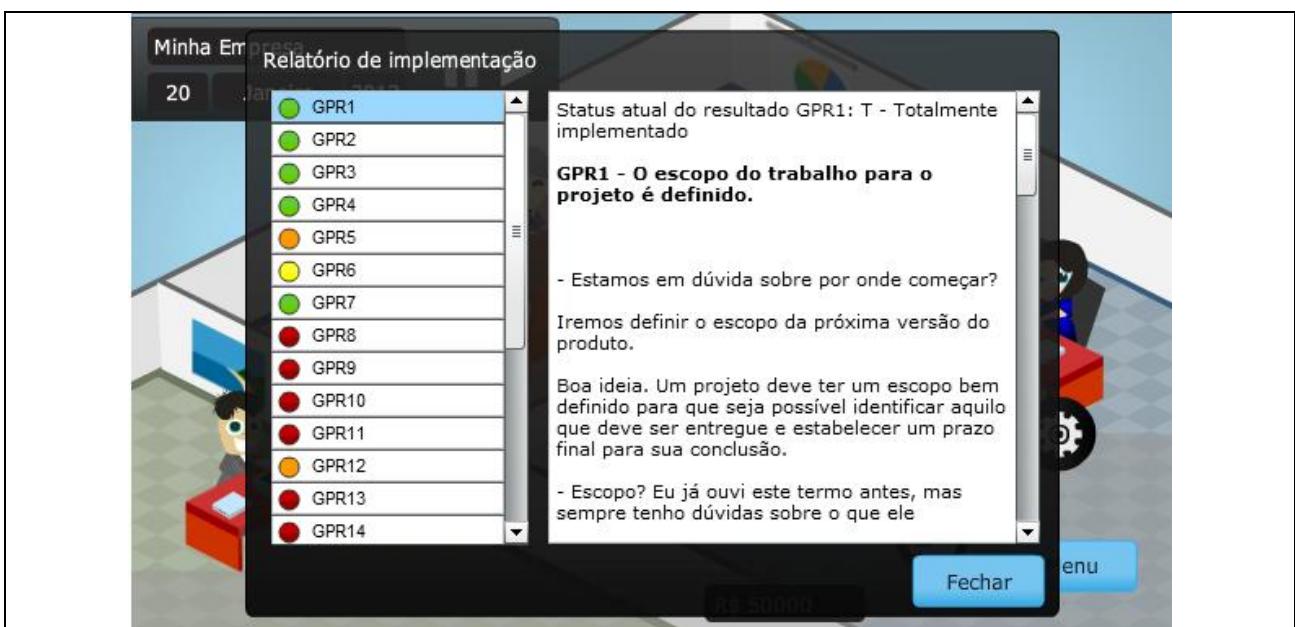


Figura 29. Tela de relatório de implementação

No setor administrativo o jogador poderá ainda consultar os gastos e receitas de cada mês. Para isso o jogador deve clicar sobre o contador e selecionar a opção “relatório de gastos e receitas”. Feito isso, será exibida a tela de relatório de gastos e receitas (Figura 30). Esta tela possui uma lista com todos os meses do ano e o jogador deve clicar sobre o mês no qual ele deseja visualizar os gastos e receitas.

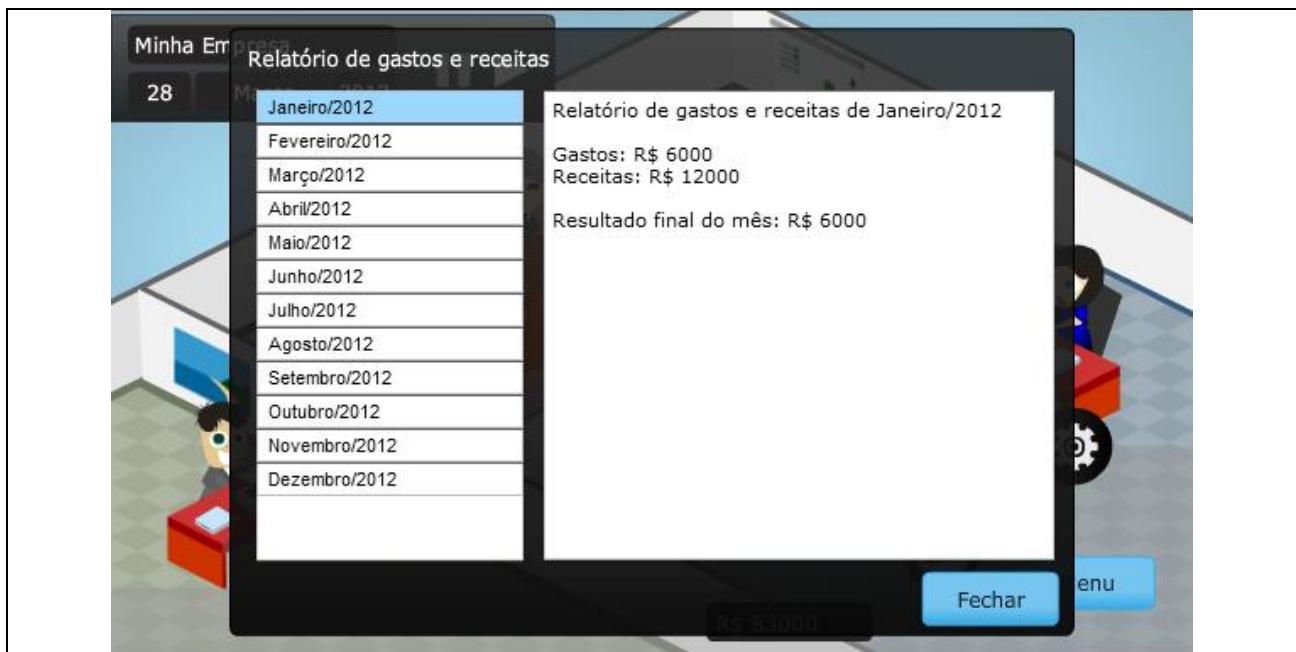


Figura 30. Tela de relatório de gastos e receitas

O jogo será concluído quando o jogador conseguir completar todas as tarefas disponíveis. Ao completar todas as tarefas disponíveis será exibida uma tela informando que o jogo foi completado (Figura 31).



Figura 31. Tela final do jogo

5 AVALIAÇÃO

Este capítulo visa apresentar o planejamento, aplicação e resultados das avaliações realizadas com o objetivo de avaliar a efetividade do jogo SPI City. Este capítulo está organizado da seguinte forma: a Seção 5.1 apresenta o planejamento dos experimentos realizados, a Seção 5.2 apresenta a avaliação qualitativa realizada na turma de especialização, a Seção 5.3 apresenta os resultados do experimento presencial, a Seção 5.4 apresenta os resultados do experimento online, e por fim a Seção 5.5 apresenta a análise dos resultados das avaliações realizadas.

5.1 PLANEJAMENTO DO EXPERIMENTO

Com o objetivo de verificar a efetividade do jogo desenvolvido segundo as hipóteses levantadas na Seção 1.1.1 foi realizado um experimento. Para auxiliar no planejamento do experimento foi utilizado o *framework* definido por Kochanski (2009), apresentado na Seção 2.2.6.1. Para o planejamento deste experimento foram instanciadas a partes I à IV do *framework*, que são apresentadas nas subseções a seguir.

5.1.1 Parte I – Definição do Experimento

A primeira parte do *framework* trata das definições iniciais de como o experimento será realizado, e é subdividida em: a) estratégia de pesquisa; b) forma de realização; c) abordagem de pesquisa; d) estratégia para seleção dos grupos; e) questionários; f) pré-condições; e g) design instrucional. A seguir cada uma destas partes, instanciadas para este experimento, serão apresentadas.

5.1.1.1 Estratégia de Pesquisa

Neste experimento foram realizadas duas avaliações distintas, uma quantitativa e outra qualitativa. Na avaliação quantitativa o objetivo era verificar a efetividade do jogo como ferramenta de ensino, para isso foram aplicados um pré-teste e um pós-teste antes e após aplicação do jogo respectivamente. Na avaliação qualitativa o objetivo era verificar a reação dos alunos quanto a aplicação do jogo, e foi utilizado como base o modelo de avaliação de jogos educacionais apresentado na Seção 2.2.6.2.

5.1.1.2 Forma de Realização

O experimento foi realizado de forma *in-vivo*, pois ele foi realizado exatamente no ambiente para o qual o jogo foi projetado, incluindo o perfil dos participantes, alunos de cursos de graduação, pós-graduação e profissionais da área de Computação/Informática e que não tiveram de ser previamente preparados para o experimento.

5.1.1.3 Abordagem de Pesquisa

Neste experimento foi utilizada a abordagem de pesquisa descritiva, pois os dados avaliados serão quantitativos e com base neste pretende-se verificar e descrever um fenômeno, estando de acordo com a regra estabelecida no *framework* para a abordagem descritiva.

5.1.1.4 Estratégia de Seleção dos Grupos

Os participantes do experimento foram aleatoriamente divididos em dois grupos (experimental e controle), sendo estes identificados, respectivamente, por grupo A e grupo B. Os grupos foram divididos da seguinte forma:

- Foram recortados papéis contendo as letras A ou B;
- Os papéis foram dobrados, embaralhados e distribuídos entre os participantes; e
- Os participantes que pegaram o papel contendo a letra A participaram do grupo A (experimental) e os participantes que pegarem o papel contendo a letra B participaram do grupo B (controle).

5.1.1.5 Questionários

Neste experimento foram utilizados os seguintes questionários:

- Pré-teste: questionário com perguntas objetivas de múltipla escolha sobre o modelo MPS.BR (Apêndice E);
- Pós-teste: mesmo questionário de pré-teste, porém as perguntas e as opções de respostas mudam de posição (Apêndice F);

- Concordância: visa obter dos participantes a formalização da concordância dos mesmos em participarem do experimento de forma voluntária (Apêndice C);
- Perfil: visa obter um conjunto de dados dos participantes que possam ajudar na análise e interpretação dos dados (Apêndice D); e
- Qualitativo: visa obter um conjunto de dados para avaliar a percepção dos alunos sobre o jogo (Apêndice G).

5.1.1.6 Pré-condições

Para a participação do experimento o candidato deve ter conhecimentos básicos em Engenharia de Software ou melhoria de processo de software.

5.1.2 Parte II – Planejamento do Experimento

Nesta segunda parte são definidos o contexto, hipóteses, variáveis de controle, a seleção dos participantes, o design do experimento, o planejamento da instrumentalização e a avaliação da validade. Cada item será apresentado individualmente de acordo com aplicação neste experimento.

5.1.2.1 Seleção do Contexto

O estudo empírico ocorreu em um laboratório de informática, para que os participantes pudessem jogar os jogos e responder os questionários, pois todos os questionários foram disponibilizados online.

5.1.2.2 Definição das Hipóteses

Este experimento visa verificar as hipóteses de pesquisa deste trabalho, apresentadas na Seção 1.1.1 . Para cada hipótese de pesquisa deste trabalho foi elaborado uma hipótese alternativa (HA) e um hipótese nula (H0), resultando nas seguintes hipóteses:

- HA₁: As pessoas que utilizam o jogo proposto apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo (Hipótese de pesquisa 1);

- H₀₁: As pessoas que utilizam o jogo proposto não apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo (Hipótese de pesquisa 1);
- H_{A2}: As pessoas que utilizam o jogo proposto apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 2);
- H₀₂: As pessoas que utilizam o jogo proposto não apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 2);
- H_{A3}: As pessoas que utilizam o jogo proposto aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 3);
- H₀₃: As pessoas que utilizam o jogo proposto não aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 3);
- H_{A4}: As pessoas que utilizam o jogo consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software (Hipótese de pesquisa 4); e
- H₀₄: As pessoas que utilizam o jogo não consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software (Hipótese de pesquisa 4).

5.1.2.3 Variáveis de Controle

Para este experimento foram definidas as seguintes variáveis de controle:

- Variável 1: pontuação dos alunos no pré-teste e pós-teste (para avaliar as hipóteses H_{A1}, H₀₁, H_{A2}, H₀₂, H_{A3} e H₀₃); e
- Variável 2: quantidade de participantes que consideram que o jogo motiva a aprendizagem de melhoria de processo de software (para avaliar as hipóteses H_{A4} e H₀₄).

5.1.2.4 Seleção dos Participantes

Os participantes eram alunos de graduação, pós-graduação ou profissionais da área de informática, e que possuíam conhecimentos básicos em Engenharia de Software ou melhoria de processo de software. Participaram apenas aqueles que assinaram o termo de concordância.

5.1.2.5 Design de Experimento Utilizado

Neste experimento os participantes foram divididos em dois grupos de maneira aleatória, foi aplicado um pré-teste antes da aplicação do jogo e um pós-teste após aplicação do mesmo, e os resultados foram analisados estatisticamente para verificar as hipóteses de pesquisa. A Figura 32 apresenta uma representação do design de experimento utilizado.

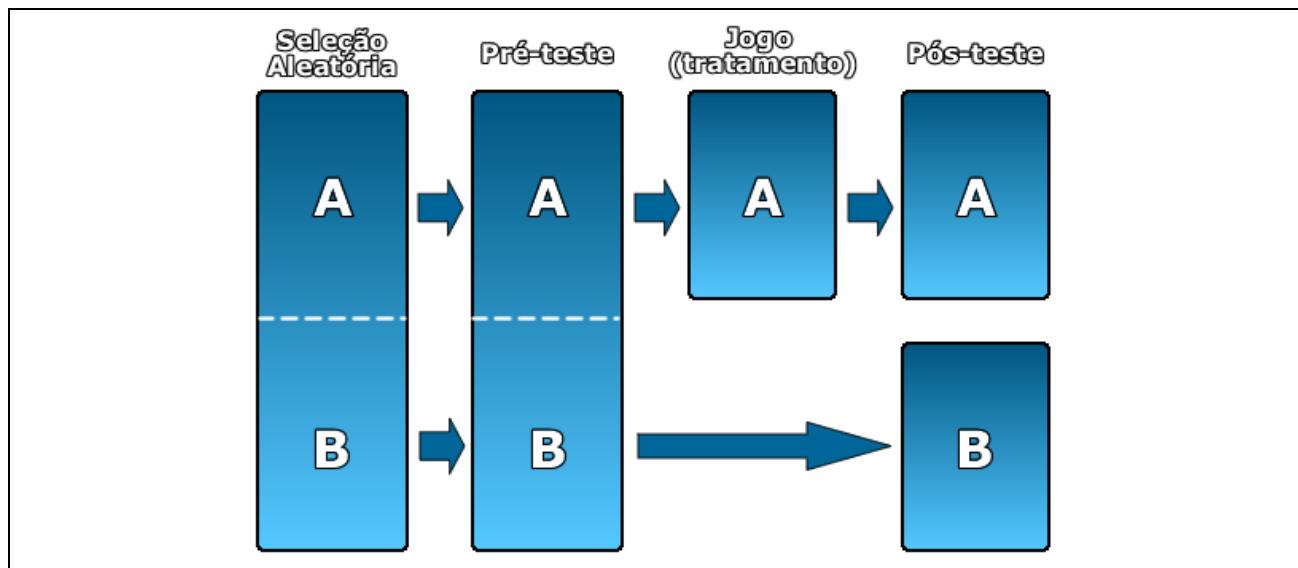


Figura 32. Design de experimento para avaliação do jogo SPI City

Fonte: Silva (2010)

Segundo Kochanski (2009), quando se usa o design de experimento de pré-teste e pós-teste, existe a ameaça de haver aprendizagem a partir das questões do pré-teste ao invés do tratamento. Para diminuir esta ameaça foi adotada a sugestão proposta pelo mesmo autor, que é a utilização das mesmas questões do pré-teste no pós-teste, mas alterando a sequencia das perguntas e das opções de respostas.

5.1.2.6 Planejamento da Instrumentação

O experimento iniciou com os participantes assinando o termo de concordância. Após assinatura do termo, os participantes responderam ao questionário de perfil, e tiveram em torno de cinco minutos para responderem este questionário. Em seguida responderam individualmente ao pré-teste, e tiveram em torno de vinte minutos para esta atividade. Após o término do pré-teste, os participantes foram divididos aleatoriamente em dois grupos (experimental e controle), conforme procedimento definido na Seção 5.1.1.4. Após a divisão dos grupos, os participantes do grupo experimental jogaram o SPI City (nome do jogo proposto) como tratamento, e os participantes do grupo de controle jogaram o jogo A Ilha dos Requisitos (GONÇALVES, 2010) como placebo², evitando que os participantes saibam se estão ou não fazendo parte do grupo experimental e por consequência recebendo o tratamento. Esta atividade teve duração de uma hora. Após o tratamento todos os alunos responderam individualmente ao pós-teste, e tiveram em torno de vinte minutos para esta atividade. Após o término do pós-teste, apenas os alunos que jogaram o SPI City responderam ao questionário qualitativo do jogo, e tiveram em torno de dez minutos para responderem a este questionário.

5.1.2.7 Avaliação da Viabilidade

Como ameaça a construção do experimento está o fato de que os participantes previamente selecionados para participação do experimento não se façam presentes no momento de aplicação ou mesmo não completem todo o ciclo do experimento. Esta ameaça foi minimizada pedindo que os participantes confirmassem sua participação e pelo fato de só participarem do experimento quem realmente queria participar.

Como ameaça de validade interna os alunos podem estudar por conta própria o assunto abordado no experimento, para reduzir as chances desta ameaça, os alunos responderam o pré-teste e o pós-teste em um mesmo dia. Porém a realização do pré-teste e pós-teste no mesmo dia pode gerar outras ameaças como, os alunos podem ficar esgotados ou podem aprender com as perguntas do pré-teste. Para minimizar a ameaça dos alunos aprenderem com o pré-teste, foram utilizados

² Técnica em que apenas os participantes não sabem sobre os detalhes do tratamento que está sendo administrado é denominada teste unicego (REIS, CICONELLI e FALOPPA, 2002, apud KOCHANSKI, 2009).

testes com as mesmas questões, mas alterando a ordem das questões e das respostas, evitando ainda que os testes possam ser incompatíveis em nível de dificuldade entre si, gerando assim outra ameaça interna.

Quanto a ameaça de validade externa, existe a possibilidade de o participante já possuir conhecimento ou experiência prévia no assunto abordado no experimento. A aplicação de pré-teste e pós-teste permite detectar estes indivíduos e se necessários podem ser eliminados do experimento.

Em relação as ameaças a conclusão do experimento está o comprometimento do pesquisador com a pesquisa e o resultado obtido. A adulteração dos resultados do pré-teste e pós-teste, pode levar a conclusões não compatíveis com a realidade. Como meio de reafirmar os resultados os quais serão obtidos nesta pesquisa, todos os detalhes do experimento são apresentadas, para que assim este possa ser replicado por outros pesquisadores.

5.1.3 Parte III – Operação do Experimento

Os participantes foram informados sobre o experimento com antecedência e todos aceitarem o termo de concordância. Os materiais que foram utilizados no experimento são os seguintes: (a) Termo de concordância; (b) Relação de participantes; (c) Questionário de perfil; (d) Pré-teste; (e) Pós-teste; (f) Jogo A Ilha dos Requisitos (GOLÇALVES, 2010); (g) Jogo SPI City; e (h) Questionário qualitativo.

As informações sobre o comprometimento dos participantes quanto ao experimento foram coletadas a partir do questionário de perfil, enquanto que os dados para verificação das hipóteses de pesquisa serão coletados a partir do pré-teste e pós-teste e do questionário qualitativo do jogo.

Quanto ao ambiente, o estudo empírico ocorreu em um laboratório de informática, para que os participantes pudessem jogar os jogos e responderem os questionários e testes, pois todos os questionários e testes foram disponibilizados online.

Com o objetivo de garantir a realização do experimento em acordo com o planejamento, todas as atividades foram acompanhadas e realizadas pelo pesquisador.

5.1.4 Parte IV – Análise e Interpretação dos Dados

Nesta parte do *framework* são apresentadas as informações para a posterior análise e interpretação dos dados, compreendendo os parâmetros populacionais, o teste a ser utilizado, a estatística que será utilizada, a forma como o conjunto de dados será reduzido e como serão conduzidos os testes das hipóteses levantadas.

5.1.4.1 Parâmetros Populacionais

Devido a dificuldade de se conseguir uma amostra de 30 alunos para participarem do experimento, principalmente considerando a divisão em dois grupos distintos, experimental e controle, foi utilizado um teste não-paramétrico, já que é independente da forma de distribuição da amostra e mesmo de uma quantidade de alunos superior ao qual se pretende aplicar.

5.1.4.2 Teste Utilizado

Para este experimento foi utilizado o teste estatístico de Mann-Whitney, pois o tipo de teste estatístico foi não-paramétrico, as amostras eram independentes, as amostras foram obtidas a partir da mesma população, a observação das amostras era independente e as observações eram comparáveis.

5.1.4.3 Estatística Utilizada

O teste de Mann-Whitney já estabelece um método para condução da verificação estatística, o qual foi seguido, conforme descrito no *framework*.

5.1.4.4 Redução do Conjunto de Dados

Para a redução do conjunto de dados foram elaborados gráficos de pontos com os dados coletados. A partir do gráfico de pontos foram analisados eventuais pontos de dispersão e tais dados foram eliminados. Caso ocorra a eliminação de algum ponto de dispersão, tal informação foi inserida nos registros do estudo.

5.1.4.5 Teste de Hipóteses

Na avaliação quantitativa foram verificadas as hipóteses HA_1 , $H0_1$, HA_2 , $H0_2$, HA_3 , e $H0_3$, tendo como base a pontuação obtida pelos alunos no pré-teste e pós-teste, em que o objetivo era verificar se o índice de conhecimento absoluto, resultante da diferença de pontuação entre o pré-teste e o pós-teste, do grupo experimental era superior ao do grupo de controle. Para esta verificação foi aplicado o teste não-paramétrico Mann-Whitney.

Na avaliação qualitativa foram verificadas as hipóteses HA_4 e $H0_4$, analisando as respostas dos alunos no questionário qualitativo utilizando a planilha desenvolvida por Savi (2011) para apoiar esta etapa. Esta planilha auxilia na organização das informações e geração automática dos gráficos com os resultados, e está disponível para download em (SAVI, 2012).

5.2 AVALIAÇÃO QUALITATIVA REALIZADA NA TURMA DE ESPECIALIZAÇÃO

No dia 14/07/2012 foi realizado uma avaliação qualitativa do jogo SPI City com alunos da disciplina de Introdução a Melhoria de Processos do curso de especialização em Engenharia e Qualidade de Software da UNIVALI, unidade de Florianópolis (SC), e contou com a participação de 20 alunos. Nesta avaliação os alunos jogaram o SPI City durante uma hora e em seguida responderam ao questionário qualitativo do jogo (Apêndice G). Os resultados desta avaliação são apresentados na Seção 5.5.2 .

5.3 EXPERIMENTO PRESENCIAL

O experimento presencial para avaliação do jogo SPI City ocorreu no dia 21/07/2012 no laboratório da UNIVALI unidade Florianópolis (SC) e contou com a participação de 11 pessoas. Todos os participantes eram graduados ou eram acadêmicos de algum curso da área de computação/informática.

O experimento foi iniciado às 9h30min com uma breve explicação sobre as etapas do mesmo, e em seguida os participantes assinaram o termo de concordância. Depois da assinatura do termo de concordância os participantes foram divididos em dois grupos: experimental e controle. Na divisão dos grupos optou-se por colocar 7 pessoas no grupo experimental e 4 no grupo de

controle, para que mais pessoas pudessem jogar o jogo objeto desse estudo. Os participantes de cada grupo foram distribuídos de maneira aleatória, através de um sorteio na qual eles retiravam de um saco um papel que indicava o grupo que eles iriam participar.

Após a divisão dos grupos, os participantes do grupo experimental foram deslocados para a parte de trás do laboratório e os participantes do grupo de controle para a parte da frente do laboratório. Essa medida foi tomada para que os alunos do grupo de controle não vissem o que os alunos do grupo experimental estavam fazendo, pois isso poderia afetar o resultado do experimento. Depois que os participantes estavam em seus devidos lugares, prosseguiu-se com a realização do experimento.

Na primeira etapa do experimento todos os participantes responderam ao questionário de perfil, esta etapa teve duração aproximada de 5 minutos. Na segunda etapa todos os participantes responderam ao pré-teste, esta etapa teve duração aproximada de 15 minutos. Na terceira etapa os participantes do grupo experimental jogaram o jogo SPI City e os participantes do grupo de controle jogaram o jogo A Ilha dos Requisitos como placebo³, sendo que esta etapa teve duração de aproximadamente 1 hora. Na quarta etapa todos os participantes responderam ao pós-teste e esta etapa teve duração aproximada de 15 minutos. Na quinta e última etapa, somente os participantes do grupo experimental responderam ao questionário de percepção do jogo, e esta etapa teve duração aproximada de 10 minutos.

5.3.1 Análise dos Resultados

Após a realização do experimento, as provas do pré-teste e do pós-teste foram corrigidas, e os dados coletados foram tabulados de forma a permitir a realização das análises e o teste das hipóteses de pesquisa. A primeira hipótese de pesquisa testada foi a hipótese H0 (o efeito de aprendizagem nos níveis de compreensão, aplicação e análise do grupo experimental não são superiores aos do grupo de controle).

Na Tabela 17 são apresentadas as notas dos participantes de cada grupo no pré-teste e no pós-teste, a diferença entre essas notas, e para preservar a identidade dos participantes, são

³ Técnica denominada unicego, em que apenas os participantes não sabem sobre os detalhes do tratamento que está sendo administrado (REIS, CICONELLI e FALOPPA, 2002, apud KOCHANSKI, 2009).

apresentados apenas os números dos participantes no experimento. De posse dos dados da Tabela 17 foi aplicado o teste estatístico Mann-Whitney⁴ para avaliação do efeito de aprendizagem relativo e efeito de aprendizagem absoluto.

Tabela 17. Notas do grupo experimental e controle no pré-teste e pós-teste

Grupo Experimental				Grupo de Controle			
Participante	Pré-teste	Pós-teste	Diferença	Participante	Pré-teste	Pós-teste	Diferença
2	2	2,5	0,5	1	1,5	0,5	-1
4	4	3	-1	3	3	2,5	-0,5
6	1	3,5	2,5	5	0,5	0,5	0
8	0,5	4,5	4	7	4	2	-2
9	3,5	3,5	0				
10	2	3,5	1,5				
11	0,5	2	1,5				

A idéia de se utilizar um jogo com foco no ensino de Engenharia de Software como placebo, era fazer com que os participantes do grupo de controle realmente não soubessem que eles não estavam recebendo o tratamento. Como o conteúdo do jogo A Ilha dos Requisitos está relacionado com os resultados esperados do processo de Gerência de Requisitos (GRE) do MPS.BR, os participantes do grupo de controle poderiam ter uma melhora nas notas do pós-teste em relação as notas do pré-teste. Mas, como pode ser observado na Tabela 17 isso não aconteceu, pois as notas dos participantes do grupo de controle não melhoraram e sim pioraram. Isso pode ter ocorrido pela falsa sensação de aprendizado causada pelo placebo, o que pode ter feito com que os participantes do grupo de controle arriscassem mais no pós-teste e consequentemente errassem mais.

5.3.1.1 Avaliação do efeito de aprendizagem relativo

Para o teste do efeito de aprendizagem relativo foram utilizadas as diferenças dos resultados do pré-teste e pós-teste de ambos os grupos. O primeiro passo foi classificar as diferenças ignorando o grupo ao qual elas pertenciam. Tal classificação está demonstrada na Tabela 18.

⁴ O teste estatístico de Mann-Whitney pode ser aplicado mesmo para amostras de tamanhos diferentes, pois a tabela de valores críticos de U corrige essa diferença (ZIMMERMAN, 1987).

Tabela 18. Classificação pelas diferenças das notas do pré e pós-teste

Sequência	Grupo	Diferença	Classificação
1	B	-2	1
2	B	-1	2,5
3	A	-1	2,5
4	B	-0,5	4
5	A	0	5,5
6	B	0	5,5
7	A	0,5	7
8	A	1,5	8,5
9	A	1,5	8,5
10	A	2,5	10
11	A	4	11

Para realizar a classificação, todos os resultados das diferenças foram colocados em ordem ascendente e a classificação iniciando de 1 foi atribuída a partir do menor resultado. No caso de haverem duas ou mais diferenças iguais, a classificação foi obtida através do cálculo da média das posições das mesmas.

No passo seguinte, os valores das classificações do grupo experimental foram calculados sob a identificação T1 e os respectivos valores do grupo de controle foram calculados sob a identificação T2. O cálculo de T1 e T2 foi obtido através da soma das classificações do grupo experimental, gerando o valor T1 e da soma das classificações do grupo de controle, gerando o valor T2. O detalhamento do cálculo é apresentado na Tabela 19.

Tabela 19. Cálculo de T1 e T2

$$T1 = 2,5 + 5,5 + 7 + 8,5 + 8,5 + 10 + 11 = 53$$

$$T2 = 1 + 2,5 + 4 + 5,5 = 13$$

Após calculados os valores de T1 e T2 foi selecionado o maior resultado. Neste caso, o resultado da maior classificação é T1 com valor 53. O valor da maior classificação será utilizado na variável Tx da fórmula do cálculo do valor de U. Os valores de n1, n2 e nx foram obtidos da seguinte forma:

- n1: número de participantes do grupo experimental;
- n2: número de participantes do grupo de controle; e
- nx: número de participantes do grupo com maior somatório de classificação.

No caso do experimento realizado o valor de n1 é igual a 7, n2 é igual a 4 e nx é igual a 7, pois havia 7 participantes no grupo experimental (n1), 4 participantes do grupo de controle (n2) e o grupo com maior somatório de classificação (nx) foi o grupo experimental com 7 participantes. De posse dos valores destas variáveis, foi realizado o cálculo de U.

$$U = n1 * n2 + nx * (nx + 1) / 2 - Tx$$

Atribuindo os valores às respectivas variáveis obtém-se:

$$U = 7 * 4 + 7 * (7 + 1) / 2 - 53$$

O valor calculado para U foi 3. O passo seguinte foi utilizar a tabela de valores críticos de U para o teste de Mann-Whitney. Na Tabela 20 de valores críticos de U, o valor da interseção entre n1 (7) e n2 (4) é 3.

Tabela 20. Tabela de valores críticos do teste U

		Nondirectional $\alpha=.05$ (Directional $\alpha=.025$)																		
		n_2																		
n_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	0	0	0	0	1	1	1	1	1	2	2	2	2
3	-	-	-	-	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
4	-	-	-	0	1	2	3	4	4	5	6	7	8	9	10	11	11	12	13	13
5	-	-	0	1	2	3	5	6	7	8	9	11	12	13	14	15	17	18	19	20
6	-	-	1	2	3	5	6	8	10	11	13	14	16	17	19	21	22	24	25	27
7	-	-	1	3	5	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
8	-	0	2	4	6	8	10	13	15	17	19	22	24	26	29	31	34	36	38	41
9	-	0	2	4	7	10	12	15	17	21	23	26	28	31	34	37	39	42	45	48
10	-	0	3	5	8	11	14	17	20	23	26	29	33	36	39	42	45	48	52	55
11	-	0	3	6	9	13	16	19	23	26	30	33	37	40	44	47	51	55	58	62
12	-	1	4	7	11	14	18	22	26	29	33	37	41	45	49	53	57	61	65	69
13	-	1	4	8	12	16	20	24	28	33	37	41	45	50	54	59	63	67	72	76
14	-	1	5	9	13	17	22	26	31	36	40	45	50	55	59	64	67	74	78	83
15	-	1	5	10	14	19	24	29	34	39	44	49	54	59	64	70	75	80	85	90
16	-	1	6	11	15	21	26	31	37	42	47	53	59	64	70	75	81	86	92	98
17	-	2	6	11	17	22	28	34	39	45	51	57	63	67	75	81	87	93	99	105
18	-	2	7	12	18	24	30	36	42	48	55	61	67	74	80	86	93	99	106	112
19	-	2	7	13	19	25	32	38	45	52	58	65	72	78	85	92	99	106	113	119
20	-	2	8	14	20	27	34	41	48	55	62	69	76	83	90	98	105	112	119	127

Fonte: <http://math.usask.ca/~laverty/S245/Tables/wmw.pdf>

Para que o resultado do teste seja significativo o valor de U obtido deve ser igual ou menor que o valor crítico obtido. Como o valor de U é igual ao valor crítico, a conclusão é a de que o efeito de aprendizagem relativo entre os grupos experimental e grupo de controle são

significativamente diferentes. Dessa forma, é possível afirmar com 95% de certeza que, para esse experimento, a hipótese H₀ não é verdadeira, o que permite aceitar a hipótese H₁, ou seja, é possível afirmar que os participantes do grupo experimental, que jogaram o jogo SPI City, tiveram um efeito de aprendizagem relativo superior aos participantes do grupo de controle.

5.3.1.2 Avaliação do efeito de Aprendizagem Absoluto

Além da avaliação realizada com enfoque no efeito de aprendizagem relativo, foi realizada a avaliação do efeito de aprendizagem absoluto. O primeiro passo foi classificar as notas pós-teste, ignorando o grupo ao qual elas pertenciam. Tal classificação está demonstrada na Tabela 21.

Tabela 21. Classificação pelas notas do pós-teste

Sequência	Grupo	Nota	Classificação
1	B	0,5	1,5
2	B	0,5	1,5
3	B	2	3,5
4	A	2	3,5
5	B	2,5	5,5
6	A	2,5	5,5
7	A	3	7
8	A	3,5	9
9	A	3,5	9
10	A	3,5	9
11	A	4,5	11

No passo seguinte, os valores das classificações do grupo experimental foram calculados sob a identificação T₁ e os respectivos valores do grupo de controle foram calculados sob a identificação T₂. O cálculo de T₁ e T₂ foi obtido através da soma das classificações do grupo experimental, gerando o valor T₁ e da soma das classificações do grupo de controle, gerando o valor T₂. O detalhamento do cálculo é apresentado na Tabela 22.

Tabela 22. Cálculo de T₁ e T₂

$T_1 = 3,5 + 5,5 + 7 + 9 + 9 + 9 + 11 = 54$
$T_2 = 1,5 + 1,5 + 3,5 + 5,5 = 12$

Após calculados os valores de T₁ e T₂ foi selecionado o maior resultado. Neste caso o resultado da maior classificação é T₁ com valor 54. O valor da maior classificação será utilizado na variável Tx da fórmula do cálculo do valor de U. Em seguida foi realizado o cálculo de U.

$$U = n1 * n2 + nx * (nx + 1) / 2 - Tx$$

Atribuindo os valores às respectivas variáveis obtém-se:

$$U = 7 * 4 + 7 * (7 + 1) / 2 - 54$$

O valor calculado para U foi 2. O passo seguinte foi utilizar a tabela de valores críticos de U para o teste de Mann-Whitney. Na Tabela 20 de valores críticos de U, o valor da interseção entre n1 (7) e n2 (4) é 3. Para que o resultado do teste seja significativo o valor de U obtido deve ser igual ou menor que o valor crítico obtido. Como o valor de U é menor que o valor crítico, a conclusão é a de que o efeito de aprendizagem absoluto entre os grupos experimental e grupo de controle são significativamente diferentes. Dessa forma, é possível afirmar com 95% de certeza que, para esse experimento, a hipótese H0 não é verdadeira, o que permite aceitar a hipótese H1, ou seja, é possível afirmar que os participantes do grupo experimental, que jogaram o jogo SPI City, tiveram um efeito de aprendizagem absoluto superior aos participantes do grupo de controle.

5.4 EXPERIMENTO ONLINE

Como poucas pessoas haviam participado do experimento presencial, decidiu-se realizar também um experimento online. Foram convidados para participar do experimento, alunos do curso de mestrado na qual essa pesquisa faz parte, alunos de especialização da área de Engenharia de Software e Qualidade de Software, e profissionais da área de informática com conhecimento em Engenharia de Software. O experimento online ficou disponível do dia 23/07/2012 ao dia 30/07/2012 e contou com a participação de somente 5 pessoas.

O experimento online contou com as mesmas etapas do experimento presencial, exceto que para este experimento não foi realizada divisão de grupos. Ao acessar o experimento era apresentado o termo de concordância de participação no experimento (Apêndice C), e uma breve descrição das etapas do experimento, indicando também o tempo necessário para realizar cada etapa. Para iniciar o experimento o participante deveria aceitar o termo de concordância. Na primeira etapa do experimento era apresentado o questionário de perfil (Apêndice D), na qual os participantes deveriam responder e depois clicar em próxima etapa para passar para segunda etapa do experimento. Na segunda etapa era apresentado pré-teste (Apêndice E), na qual os participantes deveriam responder e depois clicar em próxima etapa para passar para terceira etapa do

experimento. Na terceira etapa era apresentado o link para acesso ao jogo SPI City, nesta etapa os alunos deveriam jogar o SPI City até completá-lo, e depois deveriam voltar ao experimento e clicar em próxima etapa para ir para quarta etapa do experimento. Na quarta etapa era apresentado o pós-teste (Apêndice F), na qual os participantes deveriam responder e depois clicar em próxima etapa para ir para quinta etapa do jogo. Na quinta e última etapa do experimento era apresentado o questionário qualitativo do jogo (Apêndice G), na qual os jogadores deveriam responder para concluir o experimento.

5.4.1 Análise dos resultados

Após a realização do experimento, as provas do pré-teste e do pós-teste foram corrigidas, e os dados coletados foram tabulados de forma a permitir a realização das análises e o teste das hipóteses de pesquisa. A Tabela 23 apresentada as notas dos participantes no pré-teste e no pós-teste, apresentando a nota em cada nível da taxonomia de Bloom, e a diferença entre as notas.

Tabela 23. Notas no pré-teste e pós-teste no experimento online

Participante	Pré-teste				Pós-teste				Diferença
	CON	COMP	AP	Total	CON	COMP	AP	Total	
1	2,0	1,0	1,0	4,0	3,0	2,5	2,5	8	4,0
2	2,0	2,0	2,0	6,0	2,0	1,5	1,0	4,5	-1,5
3	1,0	2,5	1,5	5,0	2,0	2,0	1,0	5,0	0,0
4	1,5	0,5	0,0	2,0	1,5	1,5	1,0	4,0	2,0
5	1,0	1,5	1,0	3,5	1,5	3,0	1,0	5,5	2,0

Legenda: CON = Conhecimento, COMP = Compreensão, AP = Aplicação

Como neste experimento não houve divisão de grupos, para análise estatística desse experimento foi utilizado o teste de Wilcoxon (ROBSON, 2002). E para automatização dos cálculos, foi utilizado o software SPSS 13.0 para Windows, com nível de significância do teste no valor de 5 por cento. A Tabela 24 apresenta o resultado da aplicação do teste estatístico de Wilcoxon sobre os dados da Tabela 23.

Tabela 24. Resultado quantitativo do experimento online

	Conhecimento	Compreensão	Aplicação	Total
Z	-1,633 ^a	-1,125 ^a	-,552 ^a	-1,473 ^a
p-value	,102	,221	,581	,141

Legenda: a. baseado no ranking negativo.

Para que a hipótese nula seja rejeita o valor obtido no p-value deve ser menor que 0,05 (nível de significância). Como pode ser observado, em nenhum dos casos apresentados na Tabela 24 o p-value é menor que 0,05. Assim, não foi possível rejeitar as hipóteses nulas H_0_1 que afirma que as pessoas que utilizam o jogo proposto não apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam, H_0_2 que afirma que as pessoas que utilizam o jogo proposto não apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo, e H_0_3 que afirma que as pessoas que utilizam o jogo proposto não aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo.

5.5 RESULTADOS

Esta seção apresenta a análise dos resultados baseada nos dados coletados nos dois experimentos realizados e na avaliação qualitativa realizada. E está organizada da seguinte forma: a Subseção 5.5.1 apresenta a análise quantitativa dos resultados baseada em testes estatísticos realizados com intuito de avaliar as hipóteses de pesquisa, a Subseção 5.5.2 apresenta a análise qualitativa realizada com intuito de verificar a reação dos participantes sobre o jogo, a Subseção 5.5.3 apresenta a síntese dos resultados obtidos na análise quantitativa e qualitativa, e a por fim a Subseção 5.5.4 apresenta a discussão sobre as ameaças à viabilidade dos resultados.

5.5.1 Análise Quantitativa

Esta seção apresenta a análise quantitativa realizada sobre os dados coletados nos dois experimentos realizados. Para esta análise foram utilizados os resultados do pré-teste e pós-teste de todos os participantes que jogaram o jogo SPI City nos dois experimentos realizados, totalizando 12 pessoas. Para a análise estatística foi utilizado o teste de Wilcoxon (ROBSON, 2002). E para automatização dos cálculos, foi utilizado o software SPSS 13.0 para Windows, com nível de significância do teste no valor de 5 por cento. A seguir é apresentada a análise geral dos resultados (Subseção 5.5.1.1), que considera somente a nota final no pré-teste e no pós-teste, e a análise dos resultados por nível de aprendizagem (Subseção 5.5.1.2), que considera a quantidade de acertos nos níveis da taxonomia de Bloom.

5.5.1.1 Geral

A análise geral dos resultados considera somente a nota final dos participantes no pré-teste e no pós-teste. A Tabela 26 apresenta o resultado da aplicação do teste Wilcoxon sobre os dados dos 12 alunos que jogaram o jogo SPI City durante os dois experimentos realizados.

Tabela 25. Resultado quantitativo geral

Z	-2,099 ^a
p-value	,036

Legenda: a. baseado no ranking negativo.

Para que a hipótese nula seja rejeitada é preciso que o valor do p-value obtido seja menor que 0,05 (nível de significância). Como pode ser observado, o valor do p-value foi 0,036 sendo menor que 0,05, então rejeitamos a hipótese nula H_0_1 e aceitamos a hipótese alternativa HA_1 que afirma que as pessoas que utilizam o jogo proposto apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo.

5.5.1.2 Por Nível de Aprendizagem

A análise dos resultados por nível de aprendizagem considera a quantidade de acertos no pré-teste e no pós-teste para os níveis conhecimento, compreensão e aplicação da taxonomia de Bloom. A Tabela 26 apresenta o resultado da aplicação do teste Wilcoxon sobre os dados dos 12 alunos que jogaram o jogo SPI City durante os dois experimentos realizados.

Tabela 26. Análise quantitativa por nível de aprendizagem

	Conhecimento	Compreensão	Aplicação	Total
Z	-,849 ^a	-1,980 ^a	-1,513 ^a	-2,099 ^a
p-value	,396	,048	,130	,036

Legenda: a. baseado no ranking negativo.

Analizando os resultados pode-se observar que o valor de p-value é menor que 0,05 somente para o nível Compreensão, assim, rejeita-se a hipótese nula H_0_2 e aceita-se a hipótese alternativa HA_2 que afirma que as pessoas que utilizam o jogo proposto apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo. Não foi possível rejeitar a hipótese nula H_0_3 que afirma que as pessoas que utilizam o jogo proposto não

aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo.

5.5.2 Análise Qualitativa

Esta seção apresenta o resultado da análise qualitativa realizada sobre as respostas ao questionário qualitativo de todos os participantes que jogaram o jogo SPI City, durante todas as avaliações realizadas, totalizando 32 pessoas. Os resultados desta análise são apresentados a seguir, divididos nos três aspectos (motivação, experiência do usuário e aprendizagem) do modelo de avaliação proposto por Savi (2011) apresentado na Seção 2.2.6.2.

5.5.2.1 Aspecto Motivação

De modo geral, observa-se que o jogo teve um efeito positivo na motivação dos participantes, pois praticamente em todos os itens houve um nível de concordância alto com as afirmações. Analisando o gráfico de freqüências (Figura 33), pode-se perceber que exceto para dois dos itens da dimensão atenção, todos os demais itens receberam nota +1 (concordo parcialmente) ou +2 (concordo fortemente) por pelo menos 75% dos participantes. A seguir são apresentadas análises detalhadas para cada dimensão do aspecto motivação.

5.5.2.1.1 Dimensão Atenção

O design do jogo foi considerado atraente por 96,9% dos participantes, que atribuíram notas +1 (concordo parcialmente) ou +2 (concordo fortemente) a este item, o que é uma indicação de que o design do jogo está adequado. Em relação a variação no jogo, 68,7% dos participantes consideraram que a variação no jogo ajudou a mantê-los atentos, atribuindo notas +1 ou +2. Porém, 15,7% dos participantes não consideraram que a variação no jogo ajudou a mantê-los atentos, atribuindo notas -1 (discordo parcialmente), o que mostra que ainda há espaço para melhorias nesse item. O item que avalia se houve algo interessante no início do jogo não teve um nível de concordância tão alto como os demais desta dimensão, 34,4% dos participantes atribuíram nota 0 (indiferente), indicando que esse aspecto precisa ser melhorado.

5.5.2.1.2 Dimensão Relevância

Os participantes consideraram o conteúdo do jogo relevante, sendo que 81,3% concordaram fortemente com este item, além disso, 93,8% dos participantes concordaram que o conteúdo do jogo está conectado com conhecimentos que eles já possuíam, o que é uma indicação que o conteúdo do jogo está adequado. E ainda, o funcionamento do jogo está adequado ao jeito de 84,4% dos participantes aprenderem, o que é uma indicação de que a maneira como o conteúdo é ensinado no jogo está adequada.

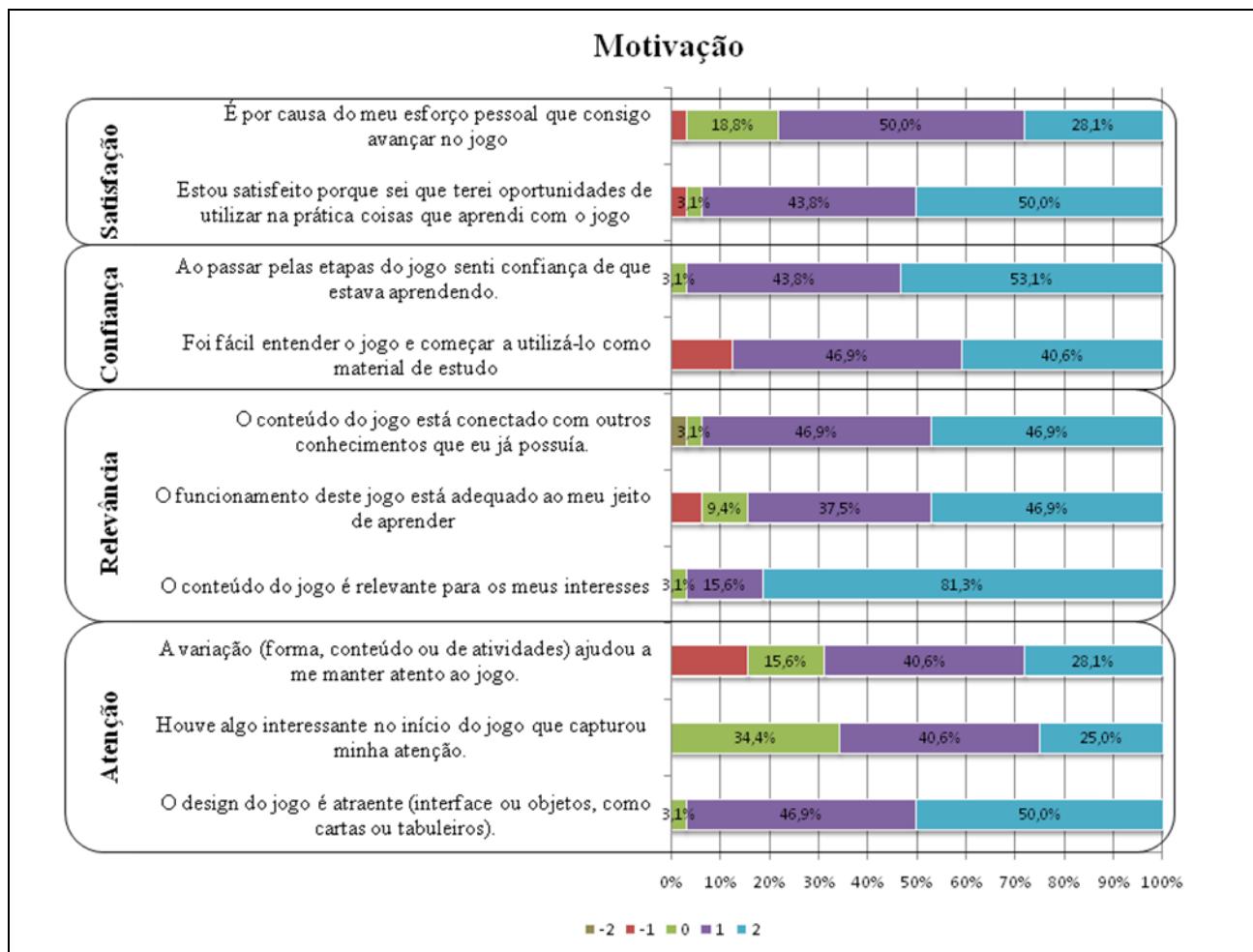


Figura 33. Aspecto motivação

5.5.2.1.3 Dimensão Confiança

Em relação a confiança, 96,9% dos participantes sentiram confiança de que estavam aprendendo ao passar pelas etapas do jogo, sendo que 53,1% concordam fortemente com este

item, o que é uma indicação de que o jogo ensina. Porém 12,5% dos participantes não consideram que o jogo é fácil de entender e de começar a utilizar, o que pode ter sido afetado pelo fato de que no momento na primeira avaliação o jogo ainda não possuía um tutorial explicando como jogar. Pode-se observar que houve redução na porcentagem de participantes que não concordam com este item em relação ao resultado da primeira avaliação, o que pode ter sido causada pela adição de um tutorial no jogo.

5.5.2.1.4 Dimensão Satisfação

Em relação a satisfação, 93,8% dos participantes ficaram satisfeitos com o jogo porque poderão utilizar na prática o que eles aprenderam no jogo. Apesar de 78,1% dos participantes concordarem que foi por causa do seu esforço pessoal que eles conseguiram avançar no jogo, somente 28,1% dos participantes concordaram fortemente com esse item, além disso, 18,8% dos participantes atribuíram nota 0 (indiferente), o que mostra que há espaço para melhorias nesse item.

5.5.2.2 Aspecto experiência do usuário

Em relação a experiência do usuário (ver Figura 34), o jogo foi considerado divertido e os participantes tiveram sentimentos de eficiência durante o jogo. Porém as dimensões imersão, interação social e desafio não foram bem avaliadas, o que indica que elas precisam ser melhoradas. A seguir a análise detalhada de cada dimensão do aspecto experiência do usuário é apresentada.

5.5.2.2.1 Dimensão imersão

A dimensão imersão teve um baixo nível de concordância pelos participantes, principalmente no item “me senti mais no ambiente do jogo do que no mundo real esquecendo temporariamente do que estava ao meu redor”, na qual 46,9% dos participantes não concordaram com esta afirmação ou consideraram como indiferente. Isso indica que os itens de imersão precisam ser melhor investigados para buscar formas de melhorar a imersão no jogo.

5.5.2.2.2 Dimensão interação social

O jogo também não foi bem avaliado nesta dimensão, pois todos os itens tiveram um nível de concordância inferior a 60%, o que indica que o jogo fornece pouca interação social entre os participantes. O item pior avaliado foi o item “O jogo promove momentos de cooperação ou

competição entre as pessoas que participam”, na qual a maioria (60%) dos participantes não concordaram com esta afirmação ou consideraram como indiferente. Em relação a esse item, está previsto para ser implementado no jogo um ranking que irá promover a competição entre os jogadores. Porém, além do ranking é preciso pensar em outras maneiras de aumentar a interação entre os jogadores.

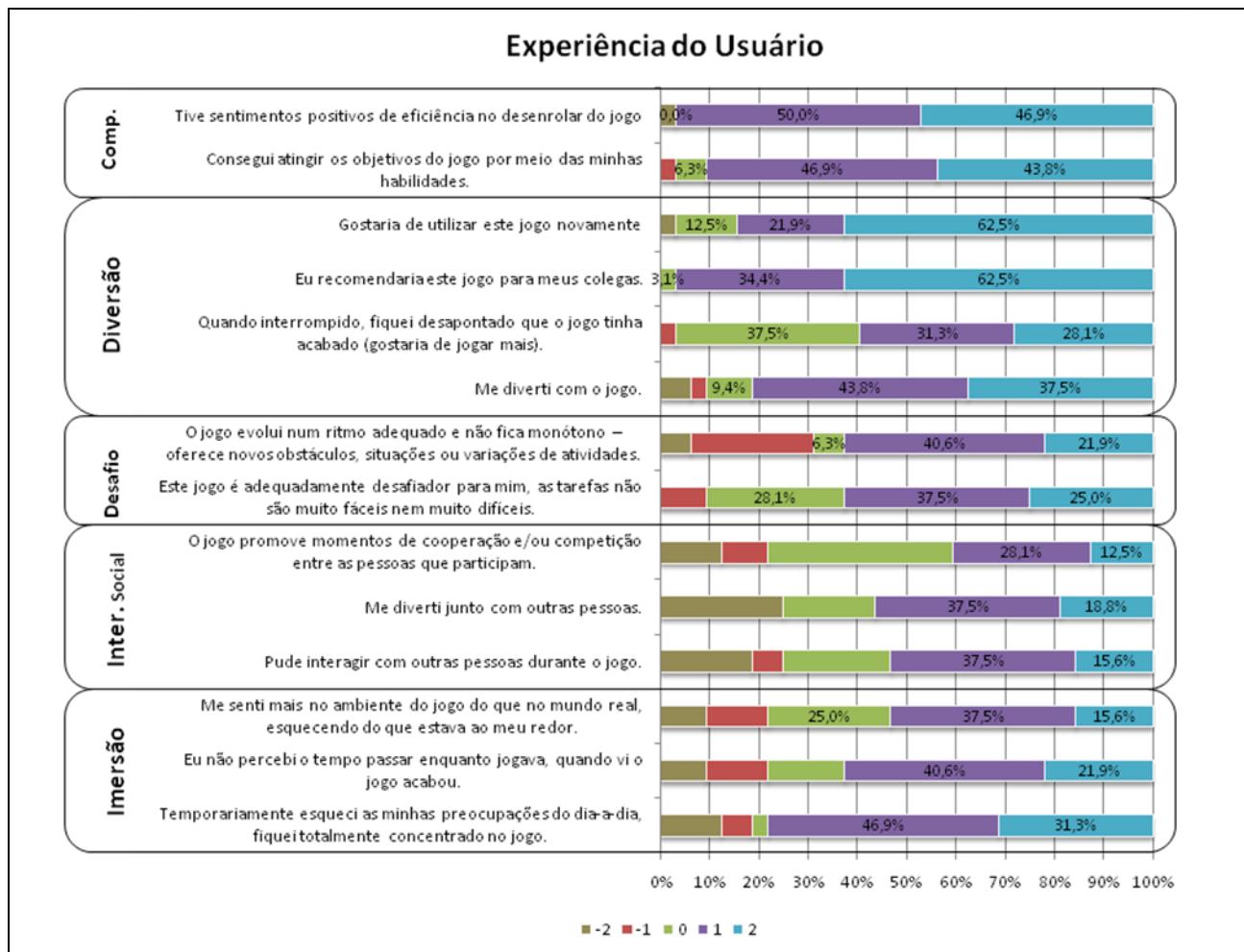


Figura 34. Aspecto experiência do usuário

5.5.2.2.3 Dimensão desafio

O jogo também teve um baixo nível de concordância nos itens dessa dimensão, principalmente no item “O jogo evolui num ritmo adequado e não fica monótono – oferece novos obstáculos, situações ou variações de atividades.”, na qual 37,5% dos participantes não concordaram com essa afirmação ou consideraram como indiferente. Isso indica que é preciso melhorar a forma como o jogo evolui para deixá-lo mais desafiador.

5.5.2.2.4 Dimensão diversão

Esta dimensão teve uma boa avaliação, pois três dos quatro itens dessa dimensão tiveram nível de concordância igual ou superior a 80%, o que indica que o jogo é divertido. O item que pergunta se os participantes ficaram desapontados com a interrupção e fim do jogo, teve um baixo nível de concordância, somente 59,4%. Isto pode ser uma indicação de que o jogo não deveria se estender por mais tempo e sua duração foi adequada.

5.5.2.2.5 Dimensão competência

Os itens que avaliam se os participantes atingiram os objetivos do jogo por meio de suas habilidades e se tiveram sentimentos positivos de eficiência no jogo, tiveram um nível de concordância igual ou superior a 90%. O que indica que o jogo promove um sentimento de competência nos jogadores, pois eles conseguem resolver os desafios através de suas habilidades.

5.5.2.3 Aspecto Aprendizagem

De maneira geral pode-se observar (ver Figura 35) que na percepção dos participantes o jogo contribui para o aprendizado de curto e longo prazo de melhoria de processo de software em nível de conhecimento, compreensão e aplicação. A seguir a análise detalhada de cada uma dessas dimensões é apresentada.

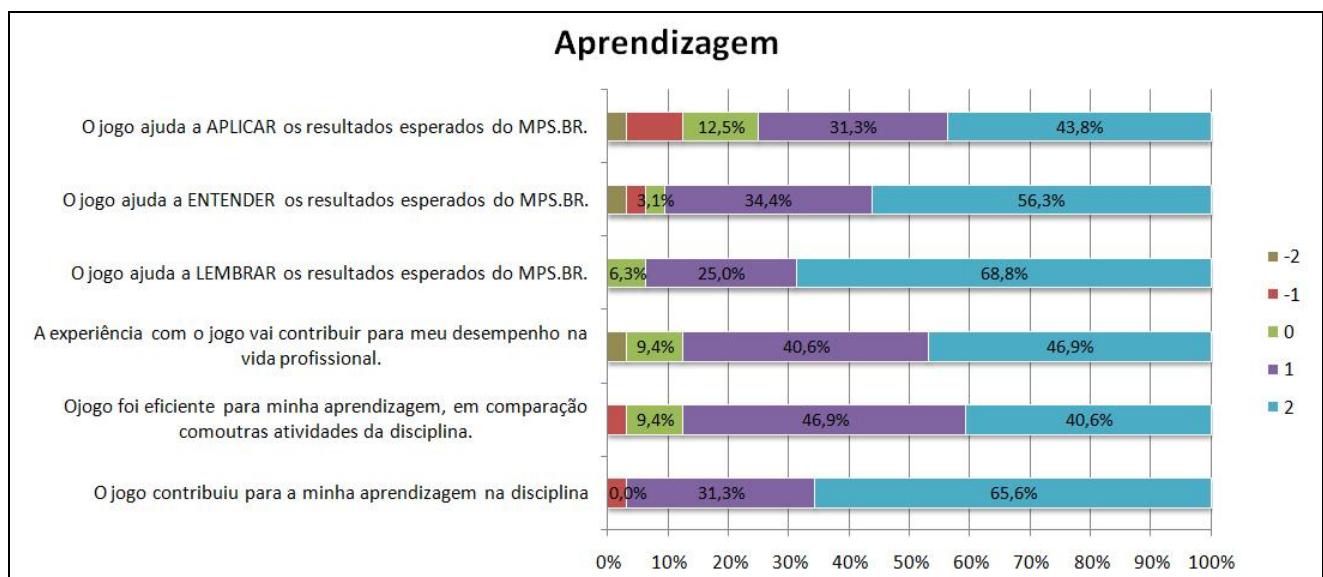


Figura 35. Aspecto aprendizagem

5.5.2.3.1 Dimensão conhecimento

Em relação a aprendizagem de nível de conhecimento, 93,3% dos participantes consideram que o jogo ajuda a lembrar os resultados esperados do MPS.BR, sendo que 68,8% dos participantes concordam fortemente com este item. O que indica que os participantes tiveram um forte sentimento que o jogo contribui para aprendizagem em nível de conhecimento de melhoria de processo de software.

5.5.2.3.2 Dimensão compreensão

A aprendizagem de nível de compreensão também teve um alto nível de concordância, 90,7% dos participantes consideram que o jogo ajuda a entender os resultados esperados do MPS.BR. Porém, o número de participantes que concordam fortemente é menor que no nível de conhecimento. Isso indica que os participantes sentem que o jogo contribui para o aprendizado em nível de compreensão de melhoria de processo de software, porém esse sentimento não é tão forte quanto para o nível de conhecimento.

5.5.2.3.3 Dimensão aplicação

A aprendizagem de aplicação teve um nível de concordância menor que na aprendizagem de conhecimento e compreensão, 75,1% dos participantes consideram que o jogo ajuda a aplicar os resultados esperados do MPS.BR. Isso indica que os participantes sentem que o jogo contribui para o aprendizado de melhoria de processo de software em nível de aplicação, porém esse sentimento não é tão forte.

5.5.2.3.4 Dimensão aprendizagem de curto termo

Em relação a aprendizagem de curto termo, 96,9% dos participantes concordam com o item que o jogo contribui para a aprendizagem na disciplina, sendo que 65,6% concordam fortemente. No item que avalia se o jogo foi eficiente para aprendizagem em comparação com outras atividades da disciplina, o nível de concordância é um pouco menor (87,5%), sendo que somente 40,6% concordam fortemente. Isso indica que os participantes sentem que o jogo contribui para aprendizagem de curto prazo, porém o sentimento da efetividade do jogo comparado a outra atividade de ensino não é tão forte.

5.5.2.3.5 Dimensão aprendizagem de longo termo

Em relação a aprendizagem de longo termo, 87,5% dos participantes concordam que a experiência no jogo vai contribuir na vida profissional, porém o número de participantes que concordam fortemente é menor do que os que concordam parcialmente. Isso indica que os participantes sentem que o jogo contribui para aprendizagem de longo prazo, mas esse sentimento não é tão forte.

5.5.2.4 Síntese dos Resultados

De acordo com o modelo de avaliação (SAVI, 2011), ao final da análise e interpretação dos resultados, a reação dos participantes sobre o jogo em relação aos aspectos motivação, experiência do usuário e aprendizagem, devem ser sumarizados em uma tabela. A Tabela 27 apresenta a reação dos participantes sobre o jogo SPI City.

Tabela 27. Síntese da reação dos participantes ao SPI City

Reação ao SPI City	Motivação	O jogo teve efeito positivo na motivação dos participantes. O design do jogo foi considerado atraente e o conteúdo relevante e conectado a outros conhecimentos. Não houve discordância que houve algo interessante no início do jogo, porém o nível de concordância foi baixo. Houve discordância de que a variação no jogo ajudou a mantê-los atentos, o que mostra que há espaço para melhorias neste item.
	Experiência do usuário	O jogo foi considerado divertido e os participantes tiveram sentimentos de eficiência durante o jogo. Houve discordância de que o jogo proporciona uma boa experiência nas dimensões imersão, interações sociais e desafio. Alguns participantes discordaram de que o jogo promove a competição entre os participantes, provavelmente porque ainda não foi implementado um ranking no jogo. É necessário melhorar a forma como o jogo evolui para deixá-lo mais desafiador. Deixando o jogo mais desafiador pode ser que aumente também o grau de imersão no jogo.
	Aprendizagem	Há percepção de que o jogo contribui para aprendizagem de curto e longo prazo de melhoria de processo de software, em nível de conhecimento, compreensão e aplicação. A percepção em relação ao nível de aplicação é menor do que para os níveis de conhecimento e compreensão.

Como observado, de acordo com a percepção dos alunos no aspecto motivação, pode-se rejeitar a hipótese nula H₀₄ e aceitar a hipótese alternativa H_{A4} que afirma que as pessoas que utilizam o jogo consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software.

5.5.3 Síntese e Análise dos Resultados

A Tabela 28 apresenta uma síntese dos resultados obtidos nesta avaliação, levando em conta as hipóteses levantadas na solução proposta (Seção 1.1.1).

Tabela 28. Síntese dos resultados da avaliação

Hipótese	Exp. 1	Exp. 2	Geral
H0 ₁ : As pessoas que utilizam o jogo proposto não apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo;		✓	
HA ₁ : As pessoas que utilizam o jogo proposto apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo	✓		✓
H0 ₂ : As pessoas que utilizam o jogo proposto não apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo	✓	✓	
HA ₂ : As pessoas que utilizam o jogo proposto apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo			✓
H0 ₃ : As pessoas que utilizam o jogo proposto não aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo	✓	✓	✓
HA ₃ : As pessoas que utilizam o jogo proposto aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo			
H0 ₄ : As pessoas que utilizam o jogo não consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software			
HA ₄ : As pessoas que utilizam o jogo consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software	✓	✓	✓

O SPI City apresentou resultados positivos analisando o resultado geral de todos os alunos que jogaram o jogo. É possível afirmar que o jogo auxilia na aprendizagem de melhoria de processo de software, principalmente em nível de compreensão.

Enquanto ao nível aplicação, não foi possível afirmar que o jogo auxilia na aprendizagem deste nível. Este fato pode ter ocorrido devido ao jogo apresentar pouca variação de conteúdo. Essa suposição é reforçada pela percepção dos participantes na dimensão atenção do aspecto motivação da avaliação qualitativa, na qual houve discordância de que a variação no jogo é adequada.

Em relação a percepção dos participantes sobre o jogo, as pessoas que utilizaram o jogo consideram que ele é uma ferramenta que motiva o aprendizado em melhoria de processo de software, além disso, os participantes tiveram sentimentos de que o jogo auxilia no ensino de

melhoria de processo de software. Porém há itens no aspecto experiência de usuário que precisam ser melhorados para deixar o jogo ainda mais atrativo, como por exemplo, a imersão e o desafio.

Com base nas respostas dos participantes ao questionário qualitativo também foi possível identificar alguns pontos fortes e oportunidades de melhoria do jogo. Os participantes destacaram como pontos fortes do jogo:

- O visual;
- A usabilidade;
- A jogabilidade;
- A relevância do conteúdo;
- A possibilidade de aplicação no mundo real;
- Os diálogos; e
- A interatividade com personagens e setores da empresa.

Como oportunidades de melhoria foram destacadas:

- Necessidade de refazer todo o diálogo quando se erra uma pergunta;
- Diminuir a quantidade de textos nos diálogos;
- Explorar mais animações;
- Utilizar recursos de som;
- Desafios alternativos, como mine-games;
- Permitir um aprofundamento no assunto, como um botão “mais” em algumas telas;
- Aumentar a interação entre os usuários, como um ranking;
- Aumentar a competitividade entre os jogadores;
- Aumentar punições para erros; e
- Permitir salvar o jogo e depois continuar de onde parou.

Algumas dessas sugestões de melhoria já foram implementadas, como a necessidade de refazer todo o diálogo quando se erra uma pergunta, e o ranking. As demais sugestões foram colocadas como trabalhos futuros.

5.5.4 Ameaças à Validade

Como ameaça a construção do experimento está o fato de que os participantes previamente selecionados para participação do experimento não se façam presentes no momento de aplicação ou mesmo não completem todo o ciclo do experimento. Esta ameaça foi minimizada pedindo que os participantes confirmassem sua participação e pelo fato de só participarem do experimento quem realmente queria participar.

Como ameaça de validade interna os participantes poderiam estudar por conta própria o assunto abordado no experimento, para reduzir as chances desta ameaça, os alunos responderam o pré-teste e o pós-teste em um mesmo dia. No experimento online foram observadas a data e a hora de resposta do pré-teste e pós-teste: se a diferença fosse superior a três horas, o participante seria excluído do experimento; participantes que respondessem somente ao pré-teste também seriam excluídos. Foram excluídos 7 participantes que haviam respondido somente ao pré-teste.

A realização do pré-teste e pós-teste no mesmo dia pode gerar outras ameaças como, os alunos podem ficar esgotados ou podem aprender com as perguntas do pré-teste. Para minimizar a ameaça dos alunos aprenderem com o pré-teste, foram utilizados testes com as mesmas questões, mas alterando a ordem das questões e das respostas, evitando ainda que os testes possam ser incompatíveis em nível de dificuldade entre si, gerando assim outra ameaça interna

Quanto à ameaça de validade externa, existe a possibilidade de o participante já possuir conhecimento ou experiência prévia no assunto abordado no experimento. A aplicação de pré-teste e pós-teste permite detectar estes indivíduos e se necessário eles seriam eliminados do experimento. Não foram identificados indivíduos que possuíssem experiência no assunto explorado pelo jogo.

Em relação às ameaças a conclusão do experimento está o comprometimento do pesquisador com a pesquisa e o resultado obtido. A adulteração dos resultados do pré-teste e pós-teste pode levar a conclusões não compatíveis com a realidade. Como meio de reafirmar os resultados os quais

foram obtidos nesta pesquisa, todos os detalhes do experimento são apresentados, para que assim este possa ser replicado por outros pesquisadores.

6 CONCLUSÕES

Muitas pesquisas têm sido feitas nos últimos anos visando identificar os fatores que afetam as iniciativas de melhoria de processo de software (ROCHA *et al.*, 2005; NIAZI, WILLSON & ZOWGHI, 2006; ROCHA & WEBER, 2008; HABIB, 2009; NIAZI, WILLSON & ZOWGHI, 2009). Destacam-se como as principais dificuldades enfrentadas pelos participantes dessas iniciativas: a falta de conhecimento sobre as técnicas básicas da Engenharia de Software (ROCHA *et al.*, 2005) e a falta de entendimento nos modelos de capacidade de processo de software (TSUKUMO *et al.*, 2006). Além disso, os cursos e treinamentos de melhoria de processo de software muitas vezes são insuficientes para a capacitação das pessoas envolvidas por que possuem o enfoque muito teórico (WANGENHEIM, HAUCK & WANGENHEIM, 2009a). A partir desta motivação, o objetivo principal deste trabalho foi desenvolver um jogo para apoiar o ensino de melhoria de processo de software.

Para alcançar esse objetivo e os objetivos específicos estabelecidos nesta dissertação, foi realizado inicialmente um estudo bibliográfico, que teve, entre suas metas, o estudo dos principais conceitos envolvidos na melhoria de processo de software, na definição e construção dos jogos e na aplicação destes como ferramentas para apoio ao processo de ensino-aprendizagem. Este estudo se apresenta como a base a definição e construção do jogo desenvolvido.

No passo seguinte foi realizado um mapeamento sistemático visando identificar como o conteúdo de melhoria de processo de software é explorado pelos jogos educacionais existentes. Como resultado pode-se perceber que o conteúdo de melhoria de processo é pouco explorado pelos jogos educacionais existentes. Com exceção ao jogo 123SPI (ADORNO, 2012), que visa ajudar a ensinar a implementar os 18 processos do nível 3 do modelo CMMI, não foi encontrado mais nenhum jogo com foco específico no ensino de um modelo de capacidade de software. Entretanto, foi observado que o conteúdo de todos os demais jogos identificados estão relacionados a pelo menos um dos resultados esperados do modelo MPS.BR. Esses jogos podem ajudar as pessoas a entenderem os conceitos relacionados com os modelos de capacidade de software, porém eles não relacionam esses conceitos com as práticas e resultados esperados desses modelos.

Em seguida, baseando-se no estudo bibliográfico e no mapeamento sistemático realizados anteriormente, deu-se início à concepção do Jogo SPI City. Primeiramente foi definida uma

metáfora que permitisse o envolvimento dos participantes com os cenários simulados. A metáfora definida foi a de um ambiente empresarial, por se tratar de um ambiente comum ao público alvo do jogo desenvolvido, permitindo aos jogadores associarem a situações vivenciadas no jogo com situações do seu dia a dia. Com a definição da metáfora o objetivo específico 1 foi alcançado.

Após a definição da metáfora os objetivos educacionais do jogo foram definidos utilizando como base a taxonomia revisada de Bloom (ANDERSON & KRATHWOHL, 2001), e compreendem os níveis Lembrar, Entender e Aplicar. Os objetivos educacionais do jogo são:

1. Lembrar e compreender os resultados esperados do processo Gerência de Requisitos do nível G de maturidade do MPS.BR;
2. Lembrar e compreender os resultados esperados do processo Gerência de Projetos do nível G de maturidade do MPS.BR;
3. Aplicar os resultados esperados do processo Gerência de Requisitos do nível G de maturidade do MPS.BR; e
4. Aplicar os resultados esperados do processo Gerência de Projetos do nível G de maturidade do MPS.BR.

Após a definição dos objetivos educacionais o projeto do jogo foi concebido. O projeto do jogo é composto por dois documentos conhecidos como *Game Design Document* (GDD) e *Technical Design Document* (TDD). O GDD apresenta as principais características do jogo proposto, como gênero, público-alvo, cenários, personagens, e outras. O TDD apresenta as características técnicas do jogo, como requisitos, regras de negócio, diagramas, e outras. Também faz parte do projeto a elaboração das situações simuladas durante o jogo. Para elaboração dessas situações utilizou-se como base o guia de implementação do nível G de maturidade do MPS.BR (SOFTEX, 2011c) e o conhecimento de especialistas no modelo. A partir da realização dessas atividades o objetivo específico 2 foi alcançado.

No próximo passo o jogo foi desenvolvido de acordo com o projeto elaborado anteriormente, e foram realizados testes buscando identificar e solucionar os erros. Com a realização desta atividade o objetivo específico 3 foi alcançado.

Após a implementação do jogo, foi iniciado o planejamento dos experimentos que o avaliaram quantitativa e qualitativamente. O planejamento utilizou como base o *framework*

desenvolvido por Kochanski (2009) e o modelo de avaliação de jogos educacionais desenvolvido por Savi (2011). Os experimentos visaram verificar as seguintes hipóteses:

1. HA₁: As pessoas que utilizam o jogo proposto apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo (Hipótese de pesquisa 1);
2. H0₁: As pessoas que utilizam o jogo proposto não apresentam melhora na aprendizagem de melhoria de processo de software em relação as pessoas que não utilizam o jogo (Hipótese de pesquisa 1);
3. HA₂: As pessoas que utilizam o jogo proposto apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 2);
4. H0₂: As pessoas que utilizam o jogo proposto não apresentam um melhor entendimento dos conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 2);
5. HA₃: As pessoas que utilizam o jogo proposto aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 3);
6. H0₃: As pessoas que utilizam o jogo proposto não aplicam melhor os conceitos de melhoria de processo de software do que as pessoas que não utilizam o jogo (Hipótese de pesquisa 3);
7. HA₄: As pessoas que utilizam o jogo consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software (Hipótese de pesquisa 4); e
8. H0₄: As pessoas que utilizam o jogo não consideram que ele é uma ferramenta que motiva o aprendizado de melhoria de processo de software (Hipótese de pesquisa 4).

Com o objetivo de comprovar ou rejeitar essas hipóteses, foram realizados dois experimentos, um presencial e outro online, e uma avaliação que envolveu somente a análise qualitativa. No experimento presencial, com base na análise dos dados realizada utilizando o teste estatístico de Mann-Whitney, foi possível rejeitar a hipótese H0₁ tanto para análise do efeito de aprendizagem relativo como absoluto, comprovando assim, a hipótese HA₁.

No experimento online não houve divisão dos participantes em grupo experimental e grupo de controle. Com isso não foi possível aplicar o teste estatístico de Mann-Whitney, sendo aplicado então o teste de Wilcoxon. Com base na análise dos dados realizada, não foi possível rejeitar as hipóteses H_0_1 , H_0_2 e H_0_3 , desta forma, as hipóteses HA_1 , HA_2 , e HA_3 , não puderam ser comprovadas.

Como base na análise quantitativa realizada sobre os dados dos dois experimentos, utilizando o teste estatístico de Wilcoxon, foi possível rejeitar as hipóteses H_0_1 e H_0_2 , comprovando assim, as hipóteses HA_1 e HA_2 . Já a hipótese H_0_3 não pode ser rejeitada, desta forma, a hipótese HA_3 , não pode ser comprovada.

De acordo com os resultados obtidos, pode-se afirmar que o SPI City auxilia na aprendizagem de melhoria de processo de software de maneira geral, principalmente em nível compreensão (nível entender da taxonomia de Bloom). Já em relação ao nível de aplicação (nível aplicar da taxonomia de Bloom), não foi possível afirmar que o jogo auxilia na aprendizagem deste nível de conhecimento. Neste ponto, considera-se que aumentar a quantidade de questões voltadas ao nível de aplicação poderão fornecer mais evidência estatística.

Em relação a avaliação qualitativa, foi realizada uma análise sobre as respostas ao questionário qualitativo de todos os participantes que jogaram o jogo SPI City, durante todas as avaliações realizadas. De acordo com a percepção dos alunos no aspecto motivação, pode-se rejeitar a hipótese nula H_0_4 , comprovando assim a hipótese HA_4 . Ou seja, os participantes consideraram que o jogo é uma ferramenta que motiva o aprendizado de melhoria de processo de software, e tiveram sentimentos de que o jogo ensina sobre este conteúdo. Porém há itens no aspecto experiência de usuário que precisam ser melhorados para deixar o jogo ainda mais atrativo, como por exemplo, a imersão e o desafio.

Desta forma, o objetivo específico 4 foi alcançado, o que ocorreu com igualmente para todos os demais objetivos específicos planejados para esta pesquisa. Ressalta-se que os resultados obtidos nela não são definitivos, mas passíveis de discussão.

6.1 CONTRIBUIÇÕES DA PESQUISA

Pode-se destacar como contribuições dessa pesquisa:

- O mapeamento sistemático realizado buscando por jogos com foco no ensino de melhoria de processo de software, pois os resultados obtidos podem ser utilizados por outros pesquisadores que estejam procurando sobre jogos com foco no ensino de melhoria de processos.
- O jogo SPI City que pode ser utilizado por iniciantes da área de melhoria de processo de software para reforçar seus conhecimentos.
- As situações elaboradas para o jogo que fornecem indícios de como os resultados esperados do MPS.BR podem ser implementados.
- O experimento projetado para a avaliação do jogo, o qual pode ser melhorado ou reproduzido em diferentes turmas para permitir futuras comparações e melhor entendimento sobre os resultados com a efetividade de aprendizagem.

Entretanto, existem possibilidades de continuação da pesquisa que são exploradas na próxima seção.

6.2 TRABALHOS FUTUROS

Este trabalho não pode ser considerado uma pesquisa fechada, oferecendo diversas possibilidades de continuação. Nesta seção, algumas destas possibilidades são discutidas em maiores detalhes:

- O jogo hoje permite o jogador escolher somente uma opção como resposta para um diálogo de uma situação. Logo, sugere-se alterar esse mecanismo para permitir que o jogador possa escolher caminhos alternativos para resolver uma mesma situação. Cada caminho pode oferecer pesos diferentes, levar a linhas de discussão diferentes e aumentar a imersão no jogo.
- Em sua versão atual, o jogo não é afetado pelas escolhas do jogador nos diferentes cenários apresentados. Neste sentido, revisar o jogo para que os resultados sejam afetados pelo histórico de decisões aumentaria a realidade do jogo, além de exigir mais dos jogadores (novos desafios).

- O jogo oferece pouca interação entre os participantes. Uma forma de aumentar a interação poderia ser através de integração com uma rede social, onde parte dos diálogos poderia ser realizada.
- O jogo não permite um aprofundamento no conteúdo. Uma forma de permitir o aprofundamento no conteúdo seria colocar um botão “mais” em algumas telas que apresente mais informações sobre o assunto que está sendo discutido, ou apresente links onde o jogador pode obter mais informações.
- O jogo não permite o jogador salvar e continuar do ponto onde parou. Para as próximas versões do jogo é importante adicionar esse recurso, pois à medida que mais diálogos forem adicionados ao jogo, pode impossibilitar o jogador concluir o mesmo dia.
- O jogo possui poucas animações e não utiliza recursos sonoros. A adição de mais animações e a utilização de recursos sonoros podem deixar o jogo ainda mais atrativo.
- Revisar os questionários de pré e pós-teste para que eles tenham mais foco em perguntas de aplicação. Nesta revisão, as questões poderiam trazer mais situações reais de empresas de desenvolvimento de software.
- Realizar novos experimentos para avaliar a efetividade de aprendizagem de forma a aumentar a força estatística dos resultados.

REFERÊNCIAS

ABES – ASSOCIAÇÃO BRASILEIRA DE EMPRESAS DE SOFTWARE. Mercado Brasileiro de Software: Panorama e Tendências - 2011. Disponível em: <<http://www.abessoftware.com.br/dados-do-setor/dados-2011>>. Acessado em: 11/11/2012.

ADORNO, K. S. V. P. Jogo educacional para aplicação dos processos do CMMI nível 2 e 3. 2010. 146 f. Exame de Qualificação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2010.

ADORNO, K. S. V. P. Jogo educacional para apoiar aplicação das áreas de processo do CMMI-DEV: níveis de maturidade 2 e 3. 2012. 195 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2012.

ANACLETO, A. Método e modelo de avaliação para melhoria de processos de software em micro e pequenas empresas. 2004. 190 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2004.

ANDERSON, L. W. KRATHWOHL, D. R. (Eds.). A taxonomy for learning, teaching, and assessing: a revision of bloom's taxonomy of educational objectives. New York: Longman, 2001.

AZEVEDO, E. Game Design Document. [S.l.], 2008. Disponível em: <http://sites.google.com/site/elinaldoazevedo/Home/DesignBible_Elinaldo.doc>. Acesso em: 10 de novembro 2012.

BADDOO, N., HALL, T. De-motivators for software process improvement: An analysis of practitioners' views, Journal of Systems and Software, v. 66, n. 1, pp. 23-33, 2003.

BAKER, A.; NAVARRO, E. O.; Hoek, A. An experimental card game for teaching software engineering processes. Journal of Systems and Software, v. 75, pg. 3-16, 2005.

BARBETTA, Pedro Alberto. REIS, Marcelo Menezes. BORNIA, Antonio Cezar. Estatística para cursos de engenharia e informática. São Paulo: Atlas, 2008.

BARRETO, A.; MONTONI, M.; SANTOS, G.; ROCHA, A. R. **Gerência de Conhecimento como Apoio para a Implantação de Processos de Software.** ProQualiti (UFLA), v. 2, n.2, p. 45-50, 2006.

BARROS, A. J. S., LEHFELD, N. A. S. **Metodologia Científica.** São Paulo: Makron Books, 2000.

BERGMANN, T. O. **Implantação do MPS.BR Nível G.** 2008. 88 f. Monografia (Bacharel) - Curso de Sistemas da Informação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2008.

BENITTI, F. B. V. MOLLÉRI, J. S. Utilização de um RPG no ensino de gerenciamento e processo de desenvolvimento de software. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO - WEI, 16., 2008, Pará. **Anais.** Pará: SBC/UFPA, 2008. p. 258-267.

CRAWFORD, C. **Chris Crawford on game design.** New Riders, 2003.

DANTAS, A. BARROS, M. WERNER, C. Treinamento experimental com jogos de simulação para gerentes de projeto de software. In: Simpósio Brasileiro de Engenharia de Software – SBES, 18., 2004, Brasília. **Anais.** Brasília: SBC/UNB, 2004. p. 23-38.

DANTAS, A. R.; BARROS, M. O.; WERNER, C. M. L. A Simulation-Based Game for Project Management Experiential Learning. In: International Conference on Software Engineering & Knowledge Engineering, 2004, **Proceedings...** Banff: International Conference on Software Engineering & Knowledge Engineering, p. 19-24, 2004.

DINIZ, L. L.. **Jogo das 7 Falhas - Um Jogo Educacional para o Apoio ao Ensino do Teste de Caixa-Preta.** 2011. 230 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Itajaí, São José, 2011.

DYBA, T., 2000, **An Instrument for measuring the key factors of success in software process improvement,** Empirical Software Engineering, v. 5, n. 4, pp. 357-390.

EL EMAM, K; DROUIN, J.; MELO, W. **SPICE – The Theory and Practice of Software Process Improvement and Capability Determination.** IEEE Computer Society, 1998.

EL-EMAM, K., GOLDENSON, D., MCCURLEY, J., et al., 2001, **Modelling the likelihood of software process improvement: An exploratory study**, Empirical Software Engineering, v. 6, n. 3, pp. 207-229.

ERTMER, P. A. NEWBY, T. J. **Behaviorism, cognitivism, constructivism**: Comparing critical features from an instructional design perspective. Performance Improvement Quarterly, 6 (4), p. 50-70. 1993.

FERNANDES, P. G.; OLIVEIRA, J. L.; MENDES, F. F.; SOUZA, A. **Resultados de Implementação Cooperada do MPS.BR**. ProQualiti (UFLA), v. 3, n.3, p. 11-18, 2007.

FERNANDES, J.; SOUZA, S. M. **PlayScrum - A Card Game to Learn The Scrum Agile Method**. In: Second International Conference on Games and Virtual Worlds for Serious Applications, 2010.

FIGUEIREDO, E.; LOBATO, C.; DIAS, K.; LEITE, J.; LUCENA, C. Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução. In: XV Workshop sobre Educação em Computação (WEI), 2007, Rio de janeiro. **Anais...** Rio de Janeiro: XXVII Congresso da SBC, p. 37-46, 2007.

FIGUEIREDO, R. T.; FIGUEIREDO, C. B. C. WarGrafos – Jogo para Auxilio na Aprendizagem da Disciplina de Teoria dos Grafos. In: X SIMPÓSIO BRASILEIRO DE GAMES E ENTRETENIMENTO DIGITAL, 2011, Salvador. **Proceedings of SBGames 2011**. Salvador: SBC, 2011.

GARCIA, G. E. **A qualidade no serviço público: um estudo de caso sobre a implantação e a continuidade de programa de gestão pela qualidade total**. Revista do centro Universitário Barão de Mauá, V. 1, n.2, jul/dez 2001. Disponível em: <<http://www.baraodemaua.br/comunicacao/publicacoes/jornal/v1n2/artigo05.html>>. Acessado em: 23/02/2012.

GIBSON, D.; ALDERICH, C.; PRENSKY, M. **Games and Simulation in Online Learning**. London: INFOSCI, 2007.

GONÇALVES, Rafael Queiroz. **Desenvolvimento de um jogo para ensino de engenharia de requisitos**. 2010. 148 f. Monografia (Graduação) - Curso de Ciência da Computação,

Departamento de Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, São José, 2010.

Hall, T, Baddoo, N, Wilson, DN and Rainer, AW. 2000a. **Optimising Software Measurement Programmes Using Practitioner Input.** In: Australian Conference on Software Measurement.Sydney, 1st-3rd November.

Hall, T, Wilson, DN and Baddoo, N. 2000b. **Towards Implementing Successful Software Inspections.** In: International Conference on Software Methods and Tools (IEEE Computer Society).Wollongoing, Australia, 6th-10th November.

HAUCK, J. C. R. Uma abordagem de modelagem de processos suportada por um guia de referência alinhado ao CMMI-DEV, MPS.BR e ISO/IEC 15504. Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2007.

HERZ, J. C. Joystick Nation: how videogames ate our quarters, won our hearts, and rewired our minds. Little, Brown & Company, 1997

HOGLE, J. G. Considering games as cognitive tools: In search of effective “Edutainment”. University of Georgia. Disponível em <<http://www.twinpinefarm.com/pdfs/games.pdf>>. Acesso em: 14/05/2010. 1996.

HUMPHREY, W. S. Characterizing the Software Process: A Maturity Framework, CMU/SEI-87-TR-1, Technical Report, Software Engineering Institute, June 1987, Disponível em: <<http://www.sei.cmu.edu>>. Acessado em: 06/05/2010. 1987.

IEEE Computer Society. SWEBOK - Guide to the Software Engineering Body of Knowledge, 2004.

ISO/IEC 15504 - The International Organization for Standardization and the International Electrotechnical Commission, **ISO/IEC 15504 - Information Technology - Process Assessment, International Standard (IS)** with five parts (ISO/IEC 15504-1, 2004; ISO/IEC 15504-2, 2003; ISO/IEC 15504-3, 2004; ISO/IEC 15504-4, 2004; ISO/IEC 15504-5, 2006], 2003.

ISO/IEC 15504-1 - The International Organization for Standardization and the International Electrotechnical Commission, **ISO/IEC 15504 - Information Technology - Process Assessment – Part 1: Concepts and Vocabulary**, 2004.

ISO/IEC 15504-2 - The International Organization for Standardization and the International Electrotechnical Commission, **ISO/IEC 15504 - Information Technology - Process Assessment – Part 2**, 2003.

ISO/IEC 15504-3 - The International Organization for Standardization and the International Electrotechnical Commission, **ISO/IEC 15504 - Information Technology - Process Assessment – Part 3: Guidance on performing an assessment**, 2004.

ISO/IEC 15504-4 - The International Organization for Standardization and the International Electrotechnical Commission, **ISO/IEC 15504 - Information Technology - Process Assessment – Part 4: Guidance on using assessment results**, 2004.

ISO/IEC 15504-5 - The International Organization for Standardization and the International Electrotechnical Commission, **ISO/IEC 15504 - Information Technology - Process Assessment – Part 5: An exemplar Process Assessment Model**, 2006.

ISO - THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION - Associação Brasileira de Normas Técnicas, NBR ISO 9000:2005 – **Sistemas de gestão da qualidade – Fundamentos e vocabulário**, Rio de Janeiro, 2005.

ISO/IEC 12207 - The International Organization for Standardization and the International Electrotechnical Commission. **ISO/IEC 12207 - Systems and software engineering– Software life cycle processes**. ISO/IEC International Standard, 2008.

KAFAI, Y. B. **The Educational Potential of Electronic Games: From Games-To-Teach to Games-To-Learn.** Conference on Playing by the Rules: The Cultural Policy Challenges of Video Games, Chicago, Illinois, 2001.

KELLER, J. M. Development and use of the arcs model of motivation design. **Journal of Instructional Development**, v. 10, n. 3, p. 2-10, 1987.

KIELING, E.; ROSA, R. **Planager – um jogo para apoio ao ensino de conceitos de gerência de projetos de software.** 2006. 68 f. Trabalho de Conclusão de Curso (Manografia) - Curso de Ciências da Computação, Universidade Católica do Rio Grande do Sul, Porto Alegre, 2006.

KITCHENHAM, B. A. **Procedures for Performing Systematic Reviews**, Tech. report TR/SE-0401, Keele University, 2004.

KOCHANSKI, D. **Um framework para apoiar a construção de experimentos na avaliação empírica de jogos educacionais.** 2009. 223f. Dissertação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2009.

LUDEWIG, J. **Models in software engineering – an introduction.** In Software and Systems Modeling. 5-14. Springer Berlin / Heidelberg. Volume 2, Number 1 / March, 2003.

MCFEELEY, B. **IDEAL: A User's Guide for Software Process Improvement.** Pittsburgh, Software Engineering Institute. , 1996.

MONTONI, M. A. **Uma Abordagem para Condução de Iniciativas de Melhoria de Processos de Software.** 2007. 146 f. Exame de Qualificação (Doutorado) - Curso de Engenharia de Sistemas e Computação, Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

MONTONI, M; CERDEIRAL, C; ZANETTI, D.; ROCHA, A. R. **Uma Abordagem para Condução de Iniciativas de Melhoria de Processos de Software.** ProQualiti (UFLA), v. 3, n.3, p. 19-24, 2007.

MONTONI, Mariano Angel. **Uma investigação sobre os fatores críticos de sucesso em iniciativas de melhoria de processos de software.** 2010. 400 f. Tese (Doutorado) - Curso de Ciências em Engenharia de Sistemas e Computação, Corpo Docente do Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia (COPPE), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

MONTONI, M. A.; ROCHA, A. R. C. **Uma Investigação sobre os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processos de Software.** In: SIMPÓSIO BRASILEIRO DE

QUALIDADE DE SOFTWARE, 10., 2011, Curitiba. **Anais...** . Curitiba: Carla Alessandra Lima Reis – Universidade Federal do Pará, Ufpa, Tayana Uchôa Conte - Universidade Federal do Amazonas, Ufam, 2011. p. 151 - 165.

NAVARRO, E. SimSE: a Software Engineering simulation environment for software process education. 2006. 321 p. Dissertation (Doctor of Philosophy in Information Computer Science) – University of California, Irvine, 2006.

NIAZI, M., WILSON, D., ZOWGHI, D., 2005a, A framework for assisting the design of effective software process improvement implementation strategies, Journal of Systems and Software, v. 78, n. 2, pp. 204-222.

NIAZI, M., WILSON, D., ZOWGHI, D., 2005b, A maturity model for the implementation of software process improvement: An empirical study, Journal of Systems and Software, v. 74, n. 2 SPEC ISS, pp. 155-172.

NIAZI, M.; WILSON D.; ZOWGHI D. Critical Success Factors for Software Process Improvement Implementation: An Empirical Study. Software Process: Improvement and Practice Journal, Vol. 11, Issue. 2, pp. 193-211, 2006.

NIAZI, M.; WILSON D.; ZOWGHI D. Critical Barriers for Software Process Improvement Implementation: An Empirical Study. APSEC - Asia-Pacific Software Engineering Conference, Batu Ferringhi, 2009. **Proceedings**, Batu Ferringhi, 2009.

NING, J.; CHEN, Z.; LIU G. PDCA Process Application in the Continuous Improvement of Software Quality. In: International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE 2010), August 24-26, 2010, Changchun, China. pages 61-65.

OMG - OBJECT MANAGEMENT GROUP, Meta Object Facility (MOF) Specification, version 1.4, OMG Specification, 358 pages, April 2002.

PASSERINO, L. M. Avaliação de jogos educativos computadorizados. In: Taller Internacional de Software Educativo, Santiago (Chile), 1998. Disponível em: <<http://www.c5.cl/tise98/html/trabajos/jogosed/>>. Acessado em: 27 de março de 2012.

PAULK, M. C.; CURTIS, B.; CHRISSIS, M. B.; WEBER C. V. **Capability Maturity Model for Software, Version 1.1**, CMU/SEI-93-TR-024, Technical Report, Software Engineering Institute, February 1993, Disponível em: <<http://www.chc-3.com/class/tr24.93.pdf>>. Acessado em: 06/05/2010. 1993.

PMI - PROJECT MANAGEMENT INSTITUTE. **Estudo de Benchmarking em Gerenciamento de Projetos Brasil 2010**, Project Management Institute – Chapters Brasileiros, Disponível em: <http://www.managerbrazil.com.br/biblioteca/benchmarking_gp_2010_geral.pdf>. Acessado em: 11/11/2012.

PRENSKY, M. **Digital Game-Based Learning**. Saint Paul: Paragon House, 2007.

PRIKLADNICKI, R.; ROSA, R.; KIELING, E. Ensino de Gerência de Projetos de Software com o Planager. In: XVIII SBIE - Simpósio Brasileiro de Informática na Educação, 2007, São Paulo. **Anais...** São Paulo: XVIII SBIE - Simpósio Brasileiro de Informática na Educação, 2007.

RABOWSKY, B. **Interactive Entertainment: A Videogame Industry Guide**. Oxnard: Radiosity Press, 2009.

RAINER, AW, HALL, T, BADDOO, N and Wilson, DN. 2001. **An Overview Of The Practitioners, Processes and Products Project**. In: 6th Annual Conference of the UK Academy of Information Systems (UKAIS2001).18th - 20th April, University of Portsmouth, UK.

RAINER, A., HALL, T., 2002, **Key success factors for implementing software process improvement: A maturity-based analysis**, Journal of Systems and Software, v. 62, n. 2, pp. 71-84.

RAINER, A., HALL, T., 2003, **A quantitative and qualitative analysis of factors affecting software processes**, Journal of Systems and Software, v. 66, n. 1, pp. 7-21.

ROBSON, Colin. **Real World Research**: A Resource for Social Scientists and Practitioner-Researchers, 2nd Edition. Wiley-Blackwell, 2002.

ROCHA, A. R. C.; MONTONI, M.; SANTOS, G.; OLIVEIRA, K.; NATALI, A. C.; MILAN MPS.BR: **lições aprendidas**. Campinas, SP : Associação para Promoção da Excelência do Software Brasileiro- SOFTEX, 2008.

ROCHA, A.; MONTONI, M.; SANTOS, G.; OLIVEIRA, K.; NATALI, A.; MIAN, P.; CONTE, T.; MAFRA, S.; BARRETO, A.; ALBUQUERQUE, A.; FIGUEIREDO, S.; SOARES, A.; BIANCHI, F.; CABRAL, R.; DIAS, A. **Fatores de Sucesso e Dificuldades na Implementação de Processos de Software Utilizando o MR-MPS e o CMMI.** ProQualiti (UFLA), v. 1, p. 13-18, 2005.

ROCHA, A. R. C.; WEBER, K. C. **MPS.BR: lições aprendidas.** Campinas, SP : Associação para Promoção da Excelência do Software Brasileiro- SOFTEX, 2008.

ROCHA, V. C. **Metodologia para implementação do MPS.BR utilizando o ambiente Webapsee.** 2009. 138 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal do Pará, Belém, 2009.

RODRIGUES, J. F.; KIRNER, T. G. Benefícios, Fatores de Sucesso e Dificuldades da Implantação do Modelo MPS.BR. SBQS – SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, Belém, 2010. **Anais**, Belém, 2010.

SALVIANO, C. F., SOUZA, E. P., DOMINONI, A. C. B., NICOLETTI, A. S., Experiência de Avaliação de Processos e Planejamento da Melhoria Utilizando a Futura Norma ISO/IEC 15504 (SPICE), **Anais do WQS'99 Workshop de Qualidade de Software do SBES'99 Simpósio Brasileiro de Engenharia de Software**, p. 1-17, Florianópolis, SC, Outubro 1999.

SALVIANO, C.F. **Uma proposta orientada a perfis de capacidade de processo para evolução da melhoria de processo de software.** Tese de doutorado pela Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 2006.

SANTANA, A. F. L. **Problemas em Iniciativas de Melhoria de Processos de Software sob a Ótica de uma Teoria de Intervenção.** 2007. 191 f. Dissertação (Mestrado) - Programa de Pós Graduação em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2007.

SAVI, R.; GRESSE VON WANGENHEIM, C.; ULBRICHT, V.R.; VANZIN, T. **Proposta de um modelo para avaliação de jogos educacionais.** RENOTE. Revista Novas Tecnologias na Educação, v. 8, n. 3, p. 1-10, 2010.

SAVI, R. Avaliação de jogos voltados para a disseminação do conhecimento. 2011. 236 f. Tese (Doutor) - Curso de Engenharia e Gestão do Conhecimento, Programa de Pós-graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2011.

SAVI, R. Modelo de avaliação de jogos educacionais. 2012. Disponível em: <<https://sites.google.com/site/savisites/avaliacao-de-jogos-educacionais>>. Acesso em: 04/08/2012.

SEI - SOFTWARE ENGINEERING INSTITUTE. Standard CMMI Appraisal Method for Process Improvement(SCAMPI) A, Version 1.2: Method Definition Document, CMU/SEI-2006-HB-002, Technical Report, Software Engineering Institute, August 2006, Disponível em: <<http://www.sei.cmu.edu>>. Acessado em: 14/04/2010. 2006.

SEI - SOFTWARE ENGINEERING INSTITUTE. Performance Results from Process Improvement. Software Engineering Institute, March 2007, Disponível em: <<http://www.sei.cmu.edu>>. Acessado em: 07/05/2010. 2007.

SEI - SOFTWARE ENGINEERING INSTITUTE. CMMI for Development: Version 1.3: CMMI-DEV. USA: SEI, 2010.

SEI - SOFTWARE ENGINEERING INSTITUTE. The IDEAL Model. Disponível em: <<http://www.sei.cmu.edu/library/assets/idealmodel.pdf>>. Acessado em: março de 2012.

SILVA, E. L. MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

SILVA, A. C. Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade. 2010. 164 f. Exame de Qualificação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2010.

SMITH, P.; RAGAN, T. J. Instructional design. 2 ed. Wiley, 1999.

SMITH, R.; GOTEL, O. Gameplay to Introduce and Reinforce Requirements Engineering Practices. In: 16th IEEE International Requirements Engineering Conference, Barcelona, Spain, IEEE Computer Society Press, pages 95-104, 2008.

SOFTEX - ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **COMUNICADO SOFTEX MPS 14/2010 - Curso de Introdução ao MPS.BR (C1-MPS.BR).** Disponível em: <http://www.softex.br/mpsbr/_cursos/curso.asp?id=3002>. Acesso em: 04/06/2010. 2010a.

SOFTEX - ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **COMUNICADO SOFTEX MPS 20/2010- Curso para Implementadores do MPS.BR (C2-MPS.BR).** Disponível em: <http://www.softex.br/mpsbr/_cursos/curso.asp?id=3061>. Acesso em: 04/06/2010. 2010b.

SOFTEX - ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **COMUNICADO SOFTEX MPS 15/2010 - Curso para Avaliadores do MPS.BR (C3-MPS.BR).** Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/COMUNICADO_15-10_Curso_C3_RIO_15a17_JUL_10%281%29.pdf>. Acesso em: 04/06/2010. 2010c.

SOFTEX – ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **MPS.BR – Modelo de Negócio, versão 2010.** ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE, Julho de 2010. Disponível em: <<http://www.softex.br>>, 2010d.

SOFTEX – ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **MPS.BR - Guia Geral, versão 2011.** ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE, Junho de 2011. Disponível em: <<http://www.softex.br>>, 2011a.

SOFTEX – ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **MPS.BR – Guia de Aquisição, versão 2011.** ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE, Outubro de 2011. Disponível em: <<http://www.softex.br>>. Acesso em: 04/06/2012. 2011b.

SOFTEX – ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS, versão 2011.** ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE, Julho de 2011. Disponível em: <<http://www.softex.br>>. Acesso em: 02/08/2011. 2011c.

SOFTEX – ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **MPS.BR – Guia de Avaliação, versão 2012.** ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE, Maio de 2012. Disponível em: <<http://www.softex.br>>. Acesso em: 14/08/2012. 2012.

SOUZA, A. S.; OLIVEIRA, J. L. Identificando Fatores Críticos para o MPS.BR através de Experiências de Implantação de Processo de Software em Goiás. *ProQualiti* (UFLA), v. 1, n.2, p. 19-26, 2005.

SWEETSER, P.; WYETH, P. GameFlow: a model for evaluating player enjoyment in games. *Comput. Entertain.*, v. 3, n. 3, p. 1-24, 2005.

THIAGARAJAN, S. 1976. “Help I Am Trapped Inside an ID Model!” *NSPI Journal* (November), 10-11.

THIRY, M., WANGENHEIM, C. G. v., ZOUCHAS, A., PICKLER, K. **Rumo ao CMMI nível 2 para Micro, Pequenas e Médias Organizações.** In: Instituto Euvaldo Lodi IEL/SC (Org.). PLATIC. 1 ed. v. 1, p. 49-79, 2007.

TRAVASSOS, G. H.; KALINOWSKI, M. **iMPS 2011 : resultados de desempenho das empresas que adotaram o modelo MPS de 2008 a 2011.** Campinas, SP: SOFTEX, 2012. 36p.

TSUKUMO, A. N.; MARTINO, W. R.; SÉRGIO, M. P.; SALVIANO, C. F. Lições aprendidas na aplicação do Método Coop-MPS para Projetos Cooperativos de Melhoria de Processo de Software com MPS.BR. *ProQualiti* (UFLA), v. 2, n. 2, p. 45-56, 2006.

WANG, H.; LIU, Y. **Research on the black start field test of shandong power grid based on PDCA cycle.** In: International Conference on Electrical and Control Engineering (ICECE 2010), 25-27 junho de 2010, pages 4144-4148.

WANGENHEIM, C. G.; HAUCK, J. C. R.; WANGENHEIN, A. Um Modelo de Treinamento para Programas de Melhoria de Processo de Software. **SBQS – SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE**, Ouro Preto, 2009. **Anais**, Ouro Preto, 2009a.

WANGENHEIM, C. G.; SAVI, R.; BORGATTO, A.; **DELIVER - An Educational Game for Teaching Earned Value Management in Computing Courses.** Information and Software Technology, Elsevier, pre-print, 2011.

WANGENHEIM, C. G. V.; THIRY, M.; KOCHANSKI, D.; STEIL, L.; SILVA, D.; LINO J. Desenvolvimento de um jogo para ensino de medição de software. SBQS – SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, Ouro Preto, 2009. **Anais**, Ouro Preto, 2009b.

WEBER, Sérgio. **ASPE / MSC: Uma Abordagem para Estabelecimento de Processos de Software em Micro e Pequenas Empresas.** Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2005.

WILSON, D.N., HALL, T., BADDOO, N., 2001, **A framework for evaluation and prediction of software process improvement success**, Journal of Systems and Software, v. 59, n. 2, pp. 135-142.

YUSOFF, A.; CROWDER, R.; GILBERT, L.; WILLS, G. A Conceptual Framework for Serious Games. In: XIX IEEE International Conference on Advanced Learning Technologies, 2009, Riga. **Proceedings of ICALT 2009**. Riga: IEEE, 2009. Pg. 21-23.

YE, E.; LIU, C.; POLACK-WAHL, J. **Enhancing Software Engineering Education Using Teaching Aids in 3-D Online Virtual Worlds.** In: 37th ASEE/IEEE Frontiers in Education Conference T1E-8, Milwaukee, WI, October 2007.

ZIMMERMAN, D. W. Comparative power of Student ttest and Mann-Whitney U test for unequal sample sizes and variances. **Journal Of Experimental Education**, Spring, p. 171-174, 1987.

ZOUCAS, A. C. **Um framework de Métodos para o desenvolvimento de modelos de capacidade de processo.** 2010. 177 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2010.

ZHU. Q; WANG, T.; TAN, S. **Adapting Game Technology to Support Software Engineering Process Teaching: From SimSE to MO-SEProcess.** In: Third International Conference on Natural Computation (ICNC 2007), icnc, vol. 5, pp.777-780, 2007.

APÊNDICE A – INFORMAÇÕES EXTRAÍDAS DOS TRABALHOS ANALISADOS NO MAPEAMENTO SISTEMÁTICO

Este apêndice apresenta as informações extraídas dos trabalhos selecionados para análise pelo mapeamento sistemático. As informações foram extraídas de acordo com a estratégia de extração dos dados definida na Seção 3.1.4 . A Tabela 29 apresenta as informações extraídas dos trabalhos analisados.

Tabela 29. Informações extraídas dos trabalhos analisados no mapeamento sistemático

ID	1
NOME	Groupthink Game
REFERENCIAS	<p>En Ye, Chang Liu, J. Polack-Wahl, “Enhancing Software Engineering Education Using Teaching Aids in 3-D Online Virtual Worlds”, 37th ASEE/IEEE Frontiers in Education Conference T1E-8, Milwaukee, WI, October 2007.</p> <p>M. Ernst, “The Groupthink Specification Exercise”, In Software Engineering Education in the Modern Age: Challenges and Possibilities, Lecture Notes in Computer Science vol. 4309, Dec. 2006, pp. 89-107.</p>
DESCRIÇÃO	Jogo multi-player desenvolvido no Second Life baseado no exercício de especificação de software Groupthink. O Groupthink é um exercício que tem objetivo de ensinar os alunos sobre especificação, mostrando sua importância, ilustrando como fazer, e enfatizando a importância do trabalho em equipe e da comunicação durante este processo. A atividade inicia com uma breve aula expositiva em especificação de software. Depois os alunos formam grupos de 7 a 10 pessoas e recebem uma especificação parcial de um sistema. Cada time deve melhorar e completar a especificação recebida e todos os membros da equipe deve concordar e entender o que foi especificado. Para avaliar a especificação elaborada cada membro da equipe deve responder uma série de perguntas sobre o comportamento do sistema, sem se comunicar com os demais membros do grupo. Diferenças entre as respostas entre membros do mesmo grupo significam erros de especificação. Depois da rodada de perguntas o instrutor apresenta os resultados e os grupos discutem sobre quais estratégias foram e não foram bem sucedidas. Para facilitar a aplicação do exercício e torná-lo mais atrativo foi desenvolvido um ambiente para aplicação do exercício no Second Life.
FORMA DE UTILIZAÇÃO	Múltiplos usuários
TIPO DO TRABALHO	Computador na internet
ÁREA DE CONHECIMENTO	Requisitos de Software.

RESULTADOS ESPERADOS	GRE1, GRE2, DRE1, DRE2, DRE3, DRE4, DRE5, DRE6, DRE7, e DRE8
ATIVIDADE DE APRENDIZAGEM	Os jogadores entram no Second Life e formam grupos sentando ao redor das mesas. Cada pergunta será mostrada em uma janela pop-up, e os jogadores respondem as perguntas clicando sobre a opção desejada na janela pop-up. Depois de todos os jogadores responderem uma pergunta, a pontuação e a estatística de escolha de cada grupo será exibida no chat do Second Life. Finalmente, a pontuação final de cada grupo e o vencedor também será anunciado no chat.
CONTEXTO DE USO	Jogo para ser utilizado como ferramenta complementar ao ensino de especificação de software em cursos de graduação em computação.
RESULTADO DE APRENDIZAGEM	Conhecimento, habilidade e atitude.
EFEITO DE APRENDIZAGEM	A maioria dos estudantes considerou o jogo um pouco útil. Os principais pontos fortes destacados são a pontuação, o trabalho em equipe, e a interatividade e comunicação providenciada pelo Second Life.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas após o jogo os estudantes responderam um survey.
ID	2
NOME	MO-SEProcess
REFERENCIAS	<p>En Ye, Chang Liu, J. Polack-Wahl, "Enhancing Software Engineering Education Using Teaching Aids in 3-D Online Virtual Worlds", 37th ASEE/IEEE Frontiers in Education Conference T1E-8, Milwaukee, WI, October 2007.</p> <p>Tao Wang, Qing Zhu, "A Software Engineering Education Game in a 3-D Online Virtual Environment," etcs, vol. 2, pp.708-710, 2009 First International Workshop on Education Technology and Computer Science, 2009.</p> <p>Qing Zhu, Tao Wang, Shenglong Tan, "Adapting Game Technology to Support Software Engineering Process Teaching: From SimSE to MO-SEProcess," icnc, vol. 5, pp.777-780, Third International Conference on Natural Computation (ICNC 2007), 2007.</p>
DESCRÍÇÃO	MO-SEProcess é um jogo multi-player 3D Online baseado no jogo SimSE (um jogo 2D) desenvolvido no SecondLife. Assim como o SimSE o MO-SEProcess tem como objetivo ensinar aos alunos os princípios do processo de desenvolvimento de software através da simulação do desenvolvimento de um projeto de software de tamanho moderado. O jogador irá atuar como um engenheiro de software, e colaborar com outros jogadores para desenvolver um sistema.
FORMA DE UTILIZAÇÃO	Múltiplos usuários
TIPO DO TRABALHO	Computador na internet
ÁREA DE CONHECIMENTO	Processo de Software
RESULTADOS ESPERADOS	GPR3
ATIVIDADE DE APRENDIZAGEM	Os jogadores entram no jogo no Second Life e escolhem o papel que iram executar, cada papel possui uma série de atividades que o jogador deve executar, os jogadores podem colaborar com os outros jogadores ajudando-os com suas atividades. O objetivo é que o time consiga entregar o produto no prazo. O

	jogadores devem seguir o processo de desenvolvimento Cascata, pois é o único suportado pelo jogo.
CONTEXTO DE USO	Jogo para ser utilizado como ferramenta complementar ao ensino de processo de software em cursos de graduação em computação.
RESULTADO DE APRENDIZAGEM	Conhecimento, habilidade e atitude.
EFEITO DE APRENDIZAGEM	A maioria dos alunos considerou o jogo um pouco útil para compreender o processo de desenvolvimento de software em uma equipe de projeto. As principais habilidades aprendidas são a comunicação e colaboração entre a equipe.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas após o jogo os estudantes responderam um survey.
ID	3
NOME	Planager
REFERENCIAS	Ricardo Rosa, Eric Kieling. Planager - Um Jogo para Apoio ao Ensino de Gerência de Projetos de Software. 2006. Trabalho de Conclusão de Curso (Graduação em Bacharelado em Sistemas de Informação) - Pontifícia Universidade Católica do Rio Grande do Sul. PRIKLADNICKI, Rafael ; ROSA, Ricardo ; KIELING, Eric . Ensino de Gerência de Projetos de Software com o Planager. In: XVIII SBIE - Simpósio Brasileiro de Informática na Educação, 2007, São Paulo. XVIII SBIE - Simpósio Brasileiro de Informática na Educação, 2007.
DESCRICAÇÃO	O Planager é um jogo single-player desenvolvido para o ensino de conceitos de gerencia de projetos. O jogo é focado em cinco processos de duas áreas de conhecimentos do PMBOK que são: gerenciamento do escopo e gerenciamento do tempo. Na área de gerência de escopo foram escolhidos os processos: definição do escopo e criar a EAP. Na área de gerência de tempo foram escolhidos: definição das atividades, sequenciamento de atividades e desenvolvimento do cronograma (com foco no cálculo do caminho crítico). O jogo possui dois módulos: o módulo de tutorial onde o aluno pode revisar os conceitos de gerência de projetos aprendidos em aula e ter uma visão de como é o jogo; e o módulo jogo onde o aluno pode praticar seus conhecimentos de uma forma interativa. O jogo possui dois tipos de usuários, o jogador e o administrador. O administrador é quem cadastra os cenários que serão explorados pelo jogador no jogo. O jogo possui cinco fases que correspondem aos cinco processos abordados pelo jogo. A primeira fase aborda a definição do escopo, a segunda a criação da EAP, a terceira a definição das atividades, a quarta o sequenciamento das atividades, e por último a quinta fase aborda o caminho crítico.
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador em ambiente local.
ÁREA DE CONHECIMENTO	Gerência de Projetos
RESULTADOS ESPERADOS	GPR1 e GPR5.
ATIVIDADE DE APRENDIZAGEM	Na primeira fase o jogador visualiza a descrição detalhada do escopo do projeto e deve responder perguntas de múltipla escolha para verificar se o jogador entendeu corretamente a descrição do escopo contido no cenário. Na segunda fase o jogador cria a estrutura analítica do projeto (EAP), ele recebe vários rótulos (alguns que fazem parte da EAP e outros que não fazem) e deve associá-los aos nodos da EAP.

	Na terceira fase o jogador recebe uma lista contendo diversas atividades (apenas algumas fazem parte do escopo definido) e deve associá-las aos pacotes de trabalho da EAP. Na quarta fase o jogador deve criar relações de dependência entre as atividades da fase anterior, associando rótulos que correspondem as atividades aos seus lugares corretos no diagrama. Na quinta fase o jogador deve descobrir o caminho crítico do projeto, ele recebe a estrutura do diagrama do caminho crítico na tela e, para cada nodo do diagrama, precisa calcular a data de início mais cedo, data de início mais tarde, data de término mais cedo, data de término mais tarde e a folga.
CONTEXTO DE USO	O jogo pode ser utilizado por disciplinas de graduação que ensinam conceitos básicos de gerenciamento de projetos e por apresentar os conceitos de forma genérica, e permitir criar qualquer cenário, é possível utilizá-la não só em projetos de desenvolvimento de software.
RESULTADO DE APRENDIZAGEM	Conhecimento e habilidade.
EFEITO DE APRENDIZAGEM	Avaliações preliminares mostraram que os alunos tiveram maior facilidade para entender os conceitos de gerência de projetos, e a estratégia de pontuação usada no jogo ajudou ao professor identificar a diferenciação de conhecimentos de cada aluno. Há necessidade de incluir maior número de cenários no jogo, aperfeiçoar a interface gráfica e validar o uso educacional com técnicas de experimentação.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas ele foi apresentado para alunos da disciplina de Gerência de Projetos de Software do curso de Bacharel em Sistemas da Informação da PUCRS.
ID	4
NOME	PlayScrum
REFERENCIAS	João M. Fernandes, Sónia M. Souza. PlayScrum - A Card Game to Learn The Scrum Agile Method. In: Second International Conference on Games and Virtual Worlds for Serious Applications, 2010.
DESCRIÇÃO	O jogo de cartas PlayScrum é um jogo de competição onde cada participante assume o papel de um Scrum Master no desenvolvimento de software que segue as práticas do Scrum. O PlayScrum pode ser jogado de 2 a 5 jogadores, e é dividido em sprints que diferem de projeto para projeto e durante cada sprint o participante deve desenvolver um número de tarefas definidas no início do jogo. O PlayScrum inclui os seguintes elementos: um tabuleiro para cada jogador, cartas de Product Backlog que determina as características do projeto, cartas de problemas jogadas para um oponente, cartas de conceitos usadas pelo jogador como um remédio para os problemas lançadas por seus oponentes, cartas de desenvolvedores correspondendo aos membros do time de desenvolvimento do jogador, artefatos que representam as tarefas a serem executadas pelo time de desenvolvimento durante o projeto, e um dado. O vencedor é o jogador que primeiro executar todas as tarefas definidas para o projeto sem erros ou o jogador que tem a maior porcentagem de tarefas sem erros no final da última iteração.
FORMA DE UTILIZAÇÃO	Múltiplos usuários
TIPO DO TRABALHO	Cartas.
ÁREA DE CONHECIMENTO	Processo de Software
RESULTADOS ESPERADOS	GPR1, GPR3, GPR5, GPR7, GPR8, GPR11, RAP3, RAP4 (Nível G), RAP5 (Até o nível F), RAP8 e RAP10 (Nível G)
ATIVIDADE DE	O jogo começa com a escolha de uma carta de Product Backlog, que determina o

APRENDIZAGEM	número de sprints do projeto (cada sprint dura 4 rodadas), o número de artefatos que devem ser implementados, a complexidade do projeto, assim como o orçamento disponível para cada jogador no projeto, e cada jogador recebe duas cartas de desenvolvedores. Durante a sua vez o jogador inicia jogando o dado para definir o número de cartas novas que ele pode coletar. Depois disso ele pode: contratar ou despedir desenvolvedores, implementar novos artefatos, verificar se seus artefatos já implementados contêm erros, concertar os erros de seus artefatos, e aplicar cartas de conceitos para corrigir problemas enviados por seus oponentes. No final de sua vez ele pode ainda receber problemas dos três jogadores que jogam após ele. O soma dos desenvolvedores e das cartas de conceitos presentes no tabuleiro do jogador não pode ultrapassar o orçamento previsto para o projeto, e o número de artefatos que o jogador pode implementar em cada rodada depende do atributo habilidade de seus desenvolvedores. Se no final de uma sprint um jogador alcançar a quantidade de artefatos corretos definidas para o projeto ele é o vencedor.
CONTEXTO DE USO	O jogo foi desenvolvido para ajudar alunos de graduação aprender o método ágil Scrum.
RESULTADO DE APRENDIZAGEM	Conhecimento.
EFEITO DE APRENDIZAGEM	Em geral, o sentimento dos estudantes em relação ao jogo foi favorável. A primeira questão “Você considera o PlayScrum divertido de jogar?” teve uma média de 4.4 em uma escala de 1 (menor) a 5 (maior). A segunda questão “Qual é o nível de dificuldade do PlayScrum?” obteve uma média de 2.5. A terceira questão “Você acha que o PlayScrum aumentou seu conhecimento de Engenharia de Software?” obteve uma média de 3.8. A quarta questão “Você ganhou novos conhecimentos sobre Engenharia de Software com o PlayScrum?” obteve uma média de 3.3. A quinta questão “O jogo ensina, em geral, o processo de Engenharia de Software?” obteve uma média de 3.6. A sexta questão “Você incluiria o PlayScrum em um curso de Engenharia de Software?” obteve uma média de 3.8. Apesar de a maioria das respostas serem favoráveis, foi percebido que estudantes com pouco conhecimento em Scrum tiveram mais dificuldade em entender o jogo e suas dinâmicas. Este ponto claramente demonstra que o jogo não deve ser utilizado isoladamente, e sim complementar os conceitos teóricos ensinados em sala de aula.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas foi realizada uma validação qualitativa e exploratória do jogo com treze estudantes de mestrado da Universidade do Minho, que jogaram o jogo e depois responderam um questionário de feedback.
ID	5
NOME	Problems and Programmers
REFERENCIAS	Alex Baker, Emily Oh Navarro, André van der Hoek: An experimental card game for teaching software engineering processes. Journal of Systems and Software 75(1-2): 3-16 (2005). Emily Oh Navarro, Alex Baker, André van der Hoek. Teaching Software Engineering Using Simulation Games, in Proceedings of the 2004 International Conference on Simulation in Education, San Diego, California, January 2003. Alex Baker, Emily Oh Navarro, André van der Hoek: Problems and Programmers: An Educational Software Engineering Card Game, in Proceedings of the Twenty-Fifth International Conference on Software Engineering, Portland, Oregon, May 2003.

	<p>Alex Baker, Emily Oh Navarro, André van der Hoek: An Experimental Card Game for Teaching Software Engineering, in Proceedings of the Sixteenth Conference on Software Engineering Education and Training, Madrid, Spain, March, 2003.</p> <p>David Carrington, Alex Baker, André van der Hoek: It's All in the Game: Teaching Software Process Concepts. Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference In Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference (2005), pp. F4G-13-F4G-18.</p>
Descrição	Problems and Programmers é um jogo educacional de cartas para o ensino de Engenharia de Software que simula o processo de software. Ele é um jogo multiplayer em que dois ou mais jogadores tentam ser o primeiro a completar um projeto hipotético de software. O jogo segue os passos do modelo de ciclo de vida Cascata, mas isto não significa que os jogadores devem completar primeiro uma fase do ciclo de vida para poderem passar para outra, e além disso é permitido que ele retorne para fases anteriores no ciclo de vida. O projeto a ser desenvolvido é definido no início do jogo, quando uma única carta de projeto é retirada e colocada no meio de todos os jogadores. A carta de projeto possui quatro atributos que são: complexidade, tamanho, qualidade e orçamento. A complexidade influencia no quanto um jogador pode avançar na fase de desenvolvimento, o tamanho define o número de cartas de código que devem estar integradas para formar a aplicação, a qualidade define o número de cartas que serão inspecionadas após todas as cartas de código estiverem integradas, e o orçamento restringe o projeto no número de desenvolvedores que podem ser contratados. O jogador que primeiro finalizar o processo de integração e conseguir que todas as cartas de código inspecionadas não possuam erro, vence o jogo.
Forma de Utilização	Múltiplos usuários
Tipo do Trabalho	Cartas.
Área de Conhecimento	Processo de Software
Resultados Esperados	GPR1, GPR3, GPR7, GPR8, GPR11, RAP3, RAP4 (Nível G), RAP5 (Até o nível F), e RAP10 (Nível G)
Atividade de Aprendizagem	Na primeira fase do jogo que é a fase de requisitos o jogador deve criar um documento de requisitos. O jogador deve criar uma coluna com cartas de requisitos e o número de cartas representa o tamanho do documento de requisitos. O jogador deve decidir quando o documento está completo baseado no tamanho e na complexidade do projeto. Em cada rodada o jogador pode escolher em pegar duas cartas novas de requisitos, uma carta nova e uma carta de substituição de requisito, ou duas cartas de substituição de requisitos. Como na fase de requisitos na fase de projeto o jogador deve criar um documento de projeto criando uma coluna para colocar cartas de projeto, em cada rodada escolhe entre adicionar novas cartas, retrabalhar em cartas existentes, ou uma mistura entre as duas opções. Na fase de implementação cada programador contratado pode executar quatro ações diferentes: programar códigos bons (custa igual a complexidade do projeto), programar um código rápido (custa metade da complexidade do projeto), inspecionar código (custa um), e corrigir erros (custa um ou mais). O custo total não pode ultrapassar o atributo habilidade de um programador. Na fase de integração o jogador pode mover todas as cartas implementadas por um programador para a coluna de integração. Os códigos são integrados programador

	por programador, mas o jogador não precisa esperar que todos os programadores terminem de codificar para começar o processo de integração. A fase de manutenção ou teste de aceitação é a fase final do jogo e só pode ser executada se a quantidade de cartas de código da coluna de integração for igual ou maior que o tamanho do projeto. Nesta fase todo código integrado é colocado virado para baixo, embaralhado, e são retiradas cartas do topo da pilha de acordo com o número do atributo qualidade do projeto. Se nenhum problema aparecer o jogador venceu o jogo, caso contrário ele pode continuar o jogo colocando todas as cartas de código acima de um programador, colocando as cartas com erros na parte de baixo da coluna.
CONTEXTO DE USO	Jogo para ser utilizado como ferramenta complementar ao ensino de processo de software em cursos de graduação em computação.
RESULTADO DE APRENDIZAGEM	Conhecimento e atitude.
EFEITO DE APRENDIZAGEM	Os resultados obtidos para o questionário numérico foram os seguintes: a pergunta “Quão divertido ele é para jogar?” obteve média de 4.1, a pergunta “Quão difícil/fácil ele é para jogar?” obteve média de 3.5, a pergunta “Quão bem ele reforça o conhecimento?” obteve média de 3.3, a pergunta “Quão bem ele ensina novo conhecimento?” obteve média de 2.6, a pergunta “Quão bem ele ensina o processo de software?” obteve média de 3.4, a pergunta “Incorporaria ele como uma parte do curso de ES?” obteve média de 3.6, a pergunta “Como uma parte opcional?” obteve média de 3.7, a pergunta “Como uma parte fundamental?” obteve média de 3.3. Esses resultados foram considerados bastante encorajadores pelos desenvolvedores desta pesquisa.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas uma avaliação inicial foi feita com 28 alunos de graduação. Os alunos foram divididos em grupos de duas pessoas, os estudantes primeiramente receberam uma introdução do jogo, depois jogaram duas vezes um contra o outro, e no final foi pedido que eles preenchessem um questionário sobre suas experiências particulares. A avaliação foi executada independente de qualquer curso particular de Engenharia de Software, pois apesar de todos os alunos terem realizado o mesmo curso, eles fizeram em trimestres diferentes com professores diferentes. Além disso, os alunos foram escolhidos aleatoriamente dentre uma série de pretendentes. O questionário possuía dois tipos de questões, oito questões de pontuação numérica, onde 1 indicava uma resposta fortemente negativa e 5 uma resposta fortemente positiva, e questões abertas.
ID	6
NOME	Re-O-Poly
REFERENCIAS	<p>Smith, Renel and Orlena Gotel, “Gameplay to Introduce and Reinforce Requirements Engineering Practices”, 16th IEEE International Requirements Engineering Conference, Barcelona, Spain, IEEE Computer Society Press. (2008).</p> <p>Smith, Renel and Orlena Gotel, “RE-O-POLY: A Game to Introduce Lightweight Requirements Engineering Good Practices”, Requirements Engineering Workshop on Education and Training (REET), New Delhi, India, October 15th, 2007.</p> <p>Smith, Renel and Orlena Gotel, “Using a Game to Introduce Lightweight Requirements Engineering”, 15th IEEE International Requirements Engineering Conference, New Delhi, India, IEEE Computer Society Press. (2007): 379-380.</p> <p>Smith, Renel and Orlena Gotel, "RE-O-Poly: A Customizable Game to Introduce and Reinforce Requirements Engineering Good Practices", 7th Annual Research Day, Friday, May 8th, 2009.</p>

DESCRIÇÃO	Re-O-Poly é um jogo de tabuleiro desenvolvido para ensinar um conjunto mais simples das práticas de Engenharia de Requisitos (ER). Ele é baseado no jogo Monopólio conhecido no Brasil como Banco Imobiliário. O jogo pode ser jogado entre duas e oito pessoas, e vence o jogo quem conseguir o maior número de SSP (Stakeholder Satisfaction Points – Pontos de Satisfação de Stakeholders). Para conseguir mais SSP os jogadores devem percorrer o tabuleiro e dependendo do local do tabuleiro em que pararem eles podem: comprar projetos, desafiar seus oponentes fazendo perguntas de Engenharia de Requisitos sobre seus projetos, pegar cartas de tarefas onde eles devem responder perguntas genéricas de Engenharia de Requisitos, ou pegar cartas de cenários.
FORMA DE UTILIZAÇÃO	Múltiplos usuários
TIPO DO TRABALHO	Tabuleiro
ÁREA DE CONHECIMENTO	Requisitos de Software
RESULTADOS ESPERADOS	GRE1, GRE2, GRE5, DRE1, DRE2, DRE3, DRE4, DRE5, DRE6, DRE7, e DRE8.
ATIVIDADE DE APRENDIZAGEM	O jogador deve posicionar sua peça, que indica sua posição no tabuleiro, na posição marcada como “Comece Aqui/Iteração”, então jogar os dados e se movimentar pelo tabuleiro no sentido horário o número de espaços indicados pela soma dos dois dados. Cada vez que o jogador parar sobre ou passar sobre “Comece Aqui/Iteração”, seja jogando os dados ou pegando uma carta, o facilitador paga para ele 200 SSP. Se o jogador parar sobre um projeto que não tem dono, ele pode comprar este projeto. Se o jogador não quiser comprar o projeto, o facilitador vende o projeto fazendo um leilão com todos os participantes do jogo. Se o participante parar sobre um projeto que tenha dono e este ainda não atingiu o status de Experiente para aquele projeto, o jogador deve desafiar o dono do projeto perguntando alguma das três perguntas listadas na carta de projeto. Se o dono do projeto não conseguir responder corretamente a pergunta ele deve pagar SSPs para o desafiador, caso consiga é o desafiador que deve pagar ao dono do projeto. Se o dono do projeto não conseguir responder duas das três perguntas do projeto ele perde o projeto, mas se ele conseguir responder as três perguntas do projeto ele se torna experiente naquele projeto e cada vez que outro jogador parar sobre o projeto deve pagar SSPs ao dono do projeto. O tabuleiro possui quatro fases de ER: Elicitação, Análise e Validação, Documentação e Gerencia de Mudança. Cada fase possui três níveis de projeto: Básico, Médio e Complexo. Se um jogador conseguir os três projetos de uma dessas fases, o jogador passa a receber o dobro de SSPs para aqueles projetos. Se um jogador conseguir os quatro projetos do mesmo nível para as quatro fases de ER, o jogador passa a receber o triplo de SSPs para estes projetos. Se a posição que o jogador para indicar que ele deve pegar uma carta de cenário ele pode ter que pagar ou receber SSPs dependendo da carta retirada. As cartas de cenário informam o que acontece quando uma boa prática de ER é implementada, neste caso o jogador receberá SSPs, ou o que acontece quando esta prática não é implementada, neste caso o jogador pagará SSPs. Se a posição do jogador indicar que ele deve pegar uma carta de tarefa, ele tem a oportunidade de ganhar SSPs, respondendo um pergunta genérica de ER presente na carta retirada. O tempo de jogo é determinado no início do jogo pelos participantes. O jogador que possuir o maior número de SSPs vence o jogo, onde para a contagem das SSPs é considerado além das SSPs que o jogador possui nas mãos, os projetos que os jogadores possuem.
CONTEXTO DE	O jogo foi desenvolvido para ajudar no ensino de um conjunto mais simples das

USO	boas práticas de Engenharia de Requisitos para Engenheiros de Requisitos inexperientes.
RESULTADO DE APRENDIZAGEM	Conhecimento.
EFEITO DE APRENDIZAGEM	As pessoas que jogaram o jogo acharam divertido, envolvente e que é uma ferramenta efetiva de ensino. Na discussão após o jogo, os participantes mostraram-se mais confortáveis com as práticas de Engenharia de Requisitos, e os testes mostraram um aumento global no conhecimento de ER dos participantes. O ambiente do jogo forçou os participantes a contextualizar a práticas de ER que haviam aprendido e a enfrentar desafios diferentes. O principal benefício parece ser a capacidade de juntar relativos novatos em ER para alavancar o conhecimento um do outro e começar um diálogo de Engenharia de Requisitos. Além disso, a jogabilidade com participantes internacionais indicaram a transferência do conceito em diferentes culturas.
PRINCIPAIS DESCOBERTAS	Para avaliar a efetividade do jogo foram executados três estudos de caso, um com estudantes de graduação dos EUA, um com alunos de graduação de Camboja e um com profissionais de TI de um grande banco dos EUA que também eram estudantes de pós-graduação. Cada sessão com um grupo de estudo teve duração de aproximadamente duas horas e incluiu uma rodada de jogo, cercado por atividades de pré e pós-jogo. No início de cada sessão, todos os participantes fizeram um teste de pré-sessão e o facilitador fez uma apresentação de slides de 30 minutos dando uma visão geral das boas práticas de ER. Foi pedido então que os membros do grupo de controle deixassem a sala e fossem ler três artigos relacionados com as boas práticas de ER. O facilitador fez uma introdução do jogo RE-O-Poly para os participantes que restaram na sala, que então jogaram o RE-O-Poly por um período máximo de uma hora sobre o controle do facilitador. Depois do jogo, o grupo de controle foi trazido de volta para a sala e ambos os grupos completaram um teste de pós-sessão. Depois os participantes que jogaram preencheram uma avaliação do jogo e da experiência de jogo através de um questionário e também participaram de um esclarecimento oral com o facilitador.
ID	7
NOME	SE-RPG
REFERENCIAS	Jefferson Seide Molleri. Utilizando o RPG como ferramenta de aprendizado para o processo de desenvolvimento de software. 2006. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Universidade do Vale do Itajaí. Orientador: Fabiane Barreto Vavassori Benitti. BENITTI, Fabiane Barreto Vavassori ; Molléri, Jefferson Seide . Utilização de um RPG no ensino de gerenciamento e processo de desenvolvimento de software. In: WEI - Workshop sobre Educação em Computação, 2008, Belém do Pará. Anais do XXVIII Congresso da SBC, 2008. p. 258-267.
DESCRIÇÃO	O SE-RPG (Software Engineering Role Playing Game) é um jogo para o ensino de Engenharia de Software com foco no processo de desenvolvimento de software e gestão de projetos. O jogo simula o ambiente de desenvolvimento de software de uma empresa fictícia com diversos personagens com quem o usuário deve interagir durante o jogo. Durante o jogo o jogador pode ou deve executar as seguintes ações: selecionar o modelo de processo (Cascata, Iterativo ou Prototipação) e linguagem de programação que será utilizada, definir a equipe, atribuir tarefas, consultar o desempenho de cada funcionário, acompanhar o andamento do projeto, contratar ou demitir funcionários, etc. O objetivo do jogo é permitir os alunos experimentarem o papel de gerente de projeto de software, as decisões e consequências, considerando alguns aspectos típicos do cotidiano de um líder de

	projeto. Também permite ao professor explorar diferentes modelos de processos, suas características, vantagens e desvantagens, favorecendo a criação de uma visão crítica em seus alunos
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador em ambiente local.
ÁREA DE CONHECIMENTO	Processo de Software
RESULTADOS ESPERADOS	GPR1, GPR3, GPR7, GPR8, RAP5 (Até o nível F), RAP7 (Até o nível F), e RAP10 (Nível G)
ATIVIDADE DE APRENDIZAGEM	No início do jogo o jogador seleciona um projeto e com base nas informações disponíveis escolhe o modelo de processo e linguagem de programação que será utilizada no desenvolvimento do projeto. Depois o jogador deve montar uma equipe para desenvolver o projeto com base em uma breve descrição das habilidades de cada personagem e no custo de cada personagem. A partir daí o jogador pode iniciar o desenvolvimento do software atribuindo tarefas aos membros de sua equipe. O jogador pode controlar o andamento do projeto, verificando o tempo gasto, o orçamento restante e o percentual de conclusão das tarefas, podendo a qualquer momento contratar ou demitir funcionários. O jogo é finalizado caso o jogador consiga concluir todas as tarefas de cada etapa do modelo de processo escolhido e tenha entregado o software para o cliente, ou caso o tempo ou custo gasto no projeto ultrapasse muito o estipulado. Ao final é realizado um breve comentário a respeito das metas do jogador e resultados obtidos, com uma pontuação determinada pelas metas de prazo, custo e escopo, e também é avaliada a escolha do modelo de processo com base nas características do projeto.
CONTEXTO DE USO	O jogo permite que os alunos da área da Engenharia de Software experimentem o papel de Gerente Projeto, mas o foco principal é permitir que os alunos criem uma visão crítica sobre os processos de desenvolvimento de software.
RESULTADO DE APRENDIZAGEM	Conhecimento e atitude.
EFEITO DE APRENDIZAGEM	Pode-se dizer que o jogo é obteve um resultado positivo quando observada a quantidade de alunos que acertaram três alternativas corretas (4 antes da demonstração e 13 após a demonstração para o modelo Cascata, 6 antes da demonstração e 10 após a demonstração para o modelo Iterativo), e ainda o resultado do teste t-Student permitiu a afirmação da hipótese que a média de acertos nas questões sobre os processos de desenvolvimento apresentou diferença significativa após o uso do SE-RPG.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas foi realizada uma validação do mesmo com 24 alunos de uma disciplina de Engenharia de Software do curso de Ciência da computação. A validação consistiu da aplicação de dois questionários, um antes e outro depois da aplicação do jogo, e na aplicação do teste t-Student para se obter a tendência da amostra. Em ambos os questionários os alunos deveriam identificar quais características estão presentes no modelo Cascata e quais características estão presentes no modelo Iterativo, onde existiam três características corretas para o modelo Cascata e quatro para o modelo Iterativo.
ID	8
NOME	SimSE
REFERENCIAS	Emily Navarro, André van der Hoek, “Multi-Site Evaluation of SimSE”, In Proceedings of the The 40th ACM Technical Symposium on Computer Science Education, Chattanooga, TN, March 2009.

Emily Navarro, André van der Hoek, "On the Role of Learning Theories in Furthering Software Engineering Education", In H.J.C. Ellis, S.A. Demurjian, and J.F. Naveda (Eds), *Software Engineering: Effective Teaching and Learning Approaches and Practices*, IGI Global, 2008.

Emily Navarro, André van der Hoek, "Comprehensive Evaluation of an Educational Software Engineering Simulation Environment", In Proceedings of the Twentieth Conference on Software Engineering Education and Training, Dublin, Ireland, July 2007.

Emily Navarro, "SimSE: A Software Engineering Simulation Environment for Software Process Education", Doctoral Dissertation, Donald Bren School of Information and Computer Sciences, University of California, Irvine, 2006.

Thomas Birkhoelzer, Emily Oh Navarro, and André van der Hoek, "Teaching by Modeling instead of by Models", In Proceedings of the 6th International Workshop on Software Process Simulation and Modeling, St. Louis, MO, May 2005.

Emily Oh Navarro and André van der Hoek, "Software Process Modeling for an Educational Software Engineering Simulation Game", In *Software Process Improvement and Practice*: 10 (3), pp. 311-325. 2004.

Emily Oh Navarro and André van der Hoek, "Design and Evaluation of an Educational Software Process Simulation Environment and Associated Model", In Proceedings of the Eighteenth Conference on Software Engineering Education and Training, Ottawa, Canada, April, 2005.

Emily Oh Navarro and André van der Hoek, "SimSE: An Interactive Simulation Game for Software Engineering Education", In Proceedings of the 7 th IASTED International Conference on Computers and Advanced Technology in Education, Kauai, Hawaii, August 2004.

Emily Oh Navarro and André van der Hoek, "Software Process Modeling for an Interactive, Graphical, Educational Software Engineering Simulation Game", In Proceedings of the 5th International Workshop on Software Process Simulation and Modeling, Edinburgh, Scotland, UK, May 2004.

Emily Oh Navarro, Alex Baker, and André van der Hoek, "Teaching Software Engineering Using Simulation Games", In Proceedings of the 2004 International Conference on Simulation in Education, San Diego, California, January 2003.

Emily Oh and André van der Hoek, "Towards Game-Based Simulation as a Method of Teaching Software Engineering", In Proceedings of the 2002 Frontiers in Education Conference, Boston, MA, November, 2002.

Emily Oh, "Teaching Software Engineering through Simulation", In Proceedings of the 2002 International Conference on Software Engineering Doctoral Symposium, Orlando, Florida, May 2002.

Emily Oh and André van der Hoek, "Teaching Software Engineering through Simulation", In Proceedings of the 2001 Workshop on Education and Training (WET), Santa Barbara, CA, July 2001.

	<p>Emily Oh and André van der Hoek, "Adapting Game Technology to Support Individual and Organizational Learning", In Proceedings of the 13th International Conference on Software Engineering and Knowledge Engineering, Buenos Aires, Argentina, June 2001.</p> <p>Emily Oh and André van der Hoek, "Challenges in Using an Economic Cost Model for Software Engineering Simulation", In Proceedings of the 3rd International Workshop on Economics-Driven Software Engineering Research, Toronto, Canada, May 2001 (reprinted in Projects & Profits, 4 (8), 43-50).</p>
Descrição	SimSE é um jogo single-player no qual o jogador assume o papel de um gerente de projetos e deve gerenciar uma equipe de desenvolvedores, a fim de completar com sucesso um projeto “virtual” de Engenharia de Software. O jogador conduz o processo contratando e demitindo funcionários, atribuindo tarefas para eles, monitorando seus progressos, comprando ferramentas, entre outras coisas. No final do jogo o jogador recebe uma pontuação indicando quanto bem ele executou o jogo, e informações adicionais que foram escondidas durante todo o jogo são reveladas para mostrar ao jogador por que ele recebeu aquela pontuação.
Forma de Utilização	Usuários individuais.
Tipo do Trabalho	Computador em ambiente local.
Área de Conhecimento	Processo de Software.
Resultados Esperados	RAP 7, GPR 3, GPR 5, GPR 7, GPR 8 e GPR 11.
Atividade de Aprendizagem	Quando o jogo começa, o jogador visualiza uma narrativa inicial que descreve os objetivos da simulação. Nesta narrativa é apresentado o projeto que o jogador deve desenvolver, assim como o orçamento e o tempo que ele tem para desenvolver este projeto. A primeira coisa que o jogador deve fazer é ir até os seus colaboradores e verificar quais são suas habilidades. Ele também deve ir até as ferramentas e verificar quais ferramentas estão disponíveis e quanto custa para adquiri-las, existem ferramentas que são gratuitas e o jogador já pode adquiri-las se desejar. Feito isso, o jogador pode começar a desenvolver o projeto atribuindo tarefas a seus colaboradores. O jogador pode atribuir tarefas a qualquer jogador, mas o ideal é que ele verifique a habilidade de cada colaborador antes de atribuir as tarefas. O jogador pode acompanhar o andamento das tarefas clicando sobre os artefatos e verificando a porcentagem de conclusão. Além disso, ele pode verificar o tempo e o dinheiro gasto no projeto até o momento clicando sobre o projeto. Baseado nestas informações ele pode tomar decisões, como comprar ferramentas ou atribuir mais funcionários para executar uma determinada tarefa. Não é possível contratar mais colaboradores, mas é possível demiti-los, reduzindo assim o gasto no projeto. O avanço no jogo é controlado manualmente pelo próprio jogador, onde ele pode avançar uma unidade de tempo ou avançar até o próximo evento clicando sobre os botões específicos. Podem ocorrer diversos eventos no jogo como um colaborador ficar doente, o cliente pedir um novo requisito, um colaborador parar de trabalhar por que está cansado, etc. O jogador continua avançando o jogo até que ele consiga completar todos os artefatos e entregue o projeto para o cliente. Ao final do jogo o jogador recebe uma pontuação indicando quanto bem ele executou o jogo, e informações adicionais que foram escondidas durante todo o jogo são reveladas para que o jogador entenda por que ele recebeu aquela pontuação.
Contexto de	O jogo para ser utilizado em conjunto a um curso de Engenharia de Software como

USO	um complemento aos métodos de enexistentes.
RESULTADO DE APRENDIZAGEM	Conhecimento.
EFEITO DE APRENDIZAGEM	<ul style="list-style-type: none"> - Os alunos que jogaram o SimSE pareceram realmente terem aprendido os conceitos para qual ele foi desenvolvido para ensinar; - Os alunos acharam jogar o SimSE uma experiência relativamente atrativa; - Fornecer aos estudantes instruções apropriadas de como jogar o SimSE é crucial; - Os alunos acharam o SimSE repetitivo quando jogado bastante tempo; - O processo de aprendizado de um jogador do SimSE envolve as teorias da descoberta do conhecimento, da aprendizagem através do erro, do construtivismo, do aprender fazendo, do aprendizado localizado, e do ARCS de Keller; e - O SimSE é mais efetivo educacionalmente quando utilizado como um componente complementar a outros métodos de ensino.
PRINCIPAIS DESCOBERTAS	A avaliação do SimSE na Universidade da Califórnia consistiu em uma série de quatro estudos: (1) um estudo inicial no qual 28 alunos jogaram o SimSE e preencheram um survey sobre sua experiência; (2) um estudo em sala de aula, onde o SimSE foi utilizado em dois trimestres como uma sessão de crédito extra em um curso de introdução a Engenharia de Software no qual os alunos tiveram que jogar o SimSE e responder um conjunto de questões; (3) um estudo comparativo realizado com 19 alunos divididos randomicamente em três grupos, onde um grupo jogou o SimSE, um realizou leituras e um assistiu a aulas expositivas, e todos os participantes responderam a um pré e pós-teste antes e depois de realizar as tarefas que lhes foi designada; (4) e um estudo profundo de observação, onde os alunos foram observados jogando o SimSE em um ambiente um para um e foram entrevistados sobre sua experiência.
ID	9
NOME	SimulES
REFERENCIAS	<p>FIGUEIREDO, Eduardo ; LOBATO, Cidiane ; DIAS, Klessis ; LEITE, Julio ; LUCENA, Carlos . Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução. In: XV Workshop sobre Educação em Computação (WEI), 2007, Rio de janeiro. co-allocado ao XXVII Congresso da SBC, 2007. p. 37-46.</p> <p>Eduardo Figueiredo, Cidiane Lobato, Klessis Dias, Julio Leite and Carlos Lucena. SimulES: Um Jogo para o Ensino de Engenharia de Software. Technical Report MCC 34/06, 42 pages. Computer Science Department, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil, 2006.</p>
DESCRIÇÃO	O SimulES é uma evolução do jogo Problems and Programmers (PnP) que visa resolver algumas limitações do PnP como: a utilização de um único processo de software, utilização de cartas muito abstratas e sem informações adicionais, ausência de um tabuleiro, ausência de um mapeamento claro entre os artefatos do jogo e os conceitos de Engenharia de Software, etc. Como no PnP, o objetivo do SimulES é que os jogadores disputem para terminar um projeto de software e o vencedor será quem implantar o projeto primeiro. Os recursos do jogo são: cartões de projeto, um tabuleiro, cartas e um dado.
FORMA DE UTILIZAÇÃO	Múltiplos usuários.
TIPO DO TRABALHO	Cartas.
ÁREA DE CONHECIMENTO	Processo de Software.
RESULTADOS	GPR 1, GPR 3, GPR 5, GPR 7, GPR 11, GRE 3, RAP 4 (Nível G), RAP 5 (Até o

ESPERADOS	nível F), e RAP10 (Nível G).
ATIVIDADE DE APRENDIZAGEM	Antes do início do jogo, é escolhida uma carta de projeto que deve ficar visível a todos os jogadores, os jogadores montam seu tabuleiro e é escolhido com o dado quem começa o jogo que deve prosseguir no sentido horário. A cada rodada, o jogador da vez lança o dado e se o número retirado for entre 1 e 3 ele terá o direito de pegar no monte de problemas e conceitos o número de cartas indicadas pelo dado, mas não poderá pegar nenhuma carta de engenheiro. Se o jogador tirar entre 4 e 6 no dado, ele irá tirar 3 cartas de problemas e conceitos e a diferença (dado-3) de engenheiros de software. As cartas podem ser guardadas na mão do jogador até que ele achar o momento oportuno para jogá-la, porém o número de cartas na mão do jogador não pode ser maior que seis, caso isso aconteça ele deve descartar cartas. As cartas de engenheiros podem ser colocadas imediatamente no tabuleiro indicando a contratação do funcionário, se o somatório de todos os funcionários contratados não ultrapassar o orçamento do projeto. A cada rodada o jogador pode pegar quantas cartas de artefatos lhe convier desde que estas estejam dentro da produtividade da sua equipe de engenheiros de software já presentes no tabuleiro. Ao final de sua jogada, o jogador pode receber cartas de problemas dos três últimos oponentes que jogaram antes dele. As cartas de problemas recebidas só terão efeito na próxima rodada, isso se o jogador não possuir uma carta de conceito em mãos que invalide a carta de problema recebida. Durante cada rodada os engenheiros de software presentes no tabuleiro tem quatro opções de tarefas que eles podem exercer que são: construir artefatos que consiste em pegar novas cartas de artefatos, inspecionar artefato que consiste em desvirar uma carta de artefato que está virada com a face para baixo, corrigir um defeito que consiste em substituir um artefato com defeito por um artefato de mesma qualidade, e integrar artefatos em um módulo que só pode ser feita quando houver artefatos suficientes no tabuleiro para compor um módulo e consiste em retirar os artefatos do tabuleiro e os colocar no monte de integração. Quando o jogador possuir todos os módulos do projeto integrados ele pode dizer que terminou o projeto. Neste momento ocorre a validação por parte do cliente e o jogador só é declarado vencedor se nos módulos verificados não for encontrado nenhum defeito.
CONTEXTO DE USO	Jogo para ser utilizado como ferramenta complementar ao ensino de processo de software em cursos de graduação em computação.
RESULTADO DE APRENDIZAGEM	Conhecimento e atitude.
EFEITO DE APRENDIZAGEM	Os alunos que jogaram o jogo acharam que ele era uma lição útil e uma experiência agradável. Além disso, a maioria dos alunos sentiu que seria valioso adicionar o jogo ao currículo de um curso de Engenharia de Software.
PRINCIPAIS DESCOBERTAS	O jogo não foi avaliado de forma sistemática, mas foi realizado uma comparação qualitativa com o jogo Problems and Programmers (PnP), o foco desta comparação está nas limitações do PnP e como essas limitações foram tratadas no SimuLES. As limitações do PnP e a solução adotada no SimuLES são as seguintes: (1) Muito amarrado a um único processo de software, no SimuLES é dada a liberdade para o jogador escolher a abordagem de desenvolvimento do seu interesse, por exemplo, caso o jogador esteja na fase de codificação ele pode voltar a fase de requisitos caso seja do seu interesse; Utilização de cartas muito abstratas e sem informações adicionais, uma solução adotada para minimizar este problema foi adicionar referências para a bibliografia relacionada ao conteúdo da carta; jogo não desperta entusiasmo dos jogadores nas primeiras rodadas, para minimizar este problema as regras foram alteradas para permitir que cartas de problemas sejam jogadas em qualquer artefato e não apenas em código; o jogo não limita o número de jogadores, mas suas regras dificultam que mais pessoas joguem, no SimuLES foi

	limitado o número de jogadores entre 4 e 8, e cada jogador recebe no máximo três problemas por rodada dos três oponentes que jogaram antes dele; ausência de um tabuleiro para organização da área de projeto do jogador, foi elaborado um tabuleiro para organizar os engenheiros de software e os artefatos pertencentes ao projeto do jogador; ausência de um mapeamento claro entre os artefatos do jogo e os conceitos de Engenharia de Software, no SimulES utilizando as referências das cartas é possível saber os conceitos associados a uma carta e os identificadores e as categorias das cartas permitem checar os conjunto de artefatos que atendem ao mesmo conceito.
ID	10
NOME	The Incredible Manager
REFERENCIAS	DANTAS, A.R. ; BARROS, Márcio de Oliveira ; WERNER, Cláudia Maria Lima . A Simulation-Based Game for Project Management Experiential Learning. In: International Conference on Software Engineering & Knowledge Engineering, 2004, Banff, Alberta. Proceedings of the International Conference on Software Engineering & Knowledge Engineering, 2004. p. 19-24. DANTAS, A.R. ; BARROS, Márcio de Oliveira ; WERNER, Cláudia Maria Lima . Treinamento Experimental com Jogos de Simulação para Gerentes de Projeto de Software. In: XVIII Simpósio Brasileiro de Engenharia de Software, 2004, Brasília, DF. Anais do XVIII Simpósio Brasileiro de Engenharia de Software, 2004. p. 23-38.
DESCRIÇÃO	The Incredible Manager é um jogo de simulação para o treinamento de gerentes de projetos de software, onde o jogador atua como um gerente no planejamento e controle de um projeto de desenvolvimento de software.
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador em ambiente local.
ÁREA DE CONHECIMENTO	Gerência de Projetos.
RESULTADOS ESPERADOS	GPR 1, GPR 2, GPR 4 (Até nível F), GPR 5, GPR 7, GPR 10, GPR 11, GPR 12, GPR 13, RAP 2, RAP 3 (Nível G), RAP 5 (Até nível F), RAP 7 (Até nível F) e RAP 10 (Nível G).
ATIVIDADE DE APRENDIZAGEM	No início do jogo é apresentado o projeto a ser gerenciado pelo jogador, com a descrição do produto a ser entregue e as restrições de qualidade, cronograma e orçamento. O jogador deve então construir um plano de projeto que contenha a definição da equipe de desenvolvedores, cronograma e organização da rede de tarefas do projeto. Para isso, o jogador deve pesquisar e contratar desenvolvedores apropriados dentre os disponíveis no banco de currículos. Uma vez definida a equipe, deve-se alocar os desenvolvedores as tarefas do projeto, conforme diagrama de redes pré-definido pelo jogo. O jogador deve também definir o esforço (em dias) que ele deseja alocar para cada tarefa, sendo que o esforço em tarefas de qualidade (inspeções) é opcional, pois o jogador pode não querer executar estas tarefas. Quando o plano de projeto estiver pronto, este deve ser enviado para a aprovação do chefe, e caso seja aprovado o jogador passa para fase de execução do projeto, caso contrário ele deve refazer o plano. Durante a execução do projeto, a rede de tarefas do projeto é executada segundo o plano de projeto aceito. O jogador deve acompanhar o andamento do projeto e pode fazer modificações no plano de projeto caso este não esteja sendo executado como esperado. O jogador pode fazer modificações no plano de projeto a qualquer momento durante a fase de execução, demitindo e contratando colaboradores,

	modificando a carga horária diária de trabalho de cada um, a duração das tarefas, e os colaboradores alocados a cada uma delas. O jogo acaba quando os recursos disponíveis para o projeto (dias e fundos) acabam sem que todas as tarefas previstas para o projeto tenham sido concluídas, neste caso o projeto foi concluído com fracasso, ou quando todas as tarefas são concluídas dentro das estimativas de recursos previstas pelo planejamento, neste caso o projeto foi concluído com sucesso.
CONTEXTO DE USO	Jogo para ser utilizado como complemento ao ensino de gerenciamento de projetos em cursos de graduação, pós-graduação ou treinamentos de empresas.
RESULTADO DE APRENDIZAGEM	Conhecimento, habilidade e atitude.
EFEITO DE APRENDIZAGEM	O processo de treinamento baseado em jogos de simulação foi considerado motivador, dinâmico, prático e divertido. Participantes apontaram como aspectos importantes a pressão psicológica e nível de dificuldade do jogo. Informaram que houve aumento em habilidade de gerenciamento; O interesse por gerenciamento aumentou depois do jogo; Treinamento baseado em jogos foi considerado bom; Os alunos consideraram o treinamento divertido ou muito divertido. Alunos reclamaram que o modelo atual é um pouco simples e incapaz de representar diversas situações e eventos inesperados.
PRINCIPAIS DESCOBERTAS	Para avaliar a adoção do jogo em um processo de treinamento foram conduzidos dois estudos de casos experimentais baseado no processo proposto por Mandl-Striegnitz, que divide o treinamento em duas sessões de simulação e duas sessões de simulação intercaladas, complementando o aprendizado que tradicionalmente seria apoiado apenas por livros-texto. O procedimento planejado para execução dos estudos foi o seguinte: a) Sessão de treinamento; b) Preenchimento do questionário quantitativo (formação, experiência); c) Primeira sessão de simulação; d) Primeira sessão de discussão; e) Segunda sessão de simulação; f) Segunda sessão de discussão; e g) Preenchimento do questionário qualitativo. O procedimento não chegou a ser completamente executado por limitações de tempo, apenas uma sessão de simulação e discussão foram executadas. O primeiro estudo foi conduzido com 7 participantes (1 aluno de doutorado e 6 alunos de mestrado) de uma turma de um curso de gerenciamento de projetos de software em um curso de pós-graduação de uma universidade federal brasileira na região sudeste do país. E o segundo estudo foi conduzido com 8 participantes (6 alunos de mestrado e 2 alunos de graduação) de um laboratório de desenvolvimento de software de uma indústria conveniada a uma universidade brasileira na região sul do país.
ID	11
NOME	X-MED
REFERENCIAS	LINO, Juliana Izabel. Proposta de um jogo educacional para a área de medição e análise de software. 2007. 243 f. Manografia (Graduação) - Curso de Sistemas da Informação, Universidade Federal de Santa Catarina, Florianópolis, 2007. GRESSE VON WANGENHEIM, C.; THIRY, M.; KOCHANSKI, D. Empirical Evaluation of an Educational Game on Software Measurement. Empirical Software Engineering, Elsevier, October 2008.
DESCRIÇÃO	O X-MED é um jogo educacional que permite realizar a simulação de um programa de medição de software com enfoque no monitoramento de projetos de acordo com o nível 2 de maturidade do CMMI-DEV v1.2, com base no GQM (Goal/Question/Metric) e PSM (Practical Software and Systems Measurement). O objetivo do jogo é reforçar conceitos de medição e ensinar a competência na

	aplicação do conhecimento de medição.
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador em ambiente local.
ÁREA DE CONHECIMENTO	Gerência de Projetos e Medição de Software.
RESULTADOS ESPERADOS	RAP 4 (A partir do Nível F), RAP 21, RAP 23, RAP 24, RAP 25, RAP 26, RAP 27, RAP 28, RAP 29, RAP 32, RAP 39, RAP 40, RAP 46, MED 1, MED 2, MED 3, MED 4, MED 5, MED 6 e MED 7.
ATIVIDADE DE APRENDIZAGEM	O jogo consiste de três fases principais, incluindo a introdução, a fase principal e a introdução. Durante o jogo, o aluno executa sequencialmente cada passo de um programa de medição, por exemplo, passo 1 – caracterização do contexto, passo 2 – identificação dos objetivos de medição, passo 3.1 – definição do abstraction sheet, entre outros. Cada passo segue a mesma sequencia, onde primeiro a descrição da tarefa é apresentada, depois informações e materiais sobre a tarefa são apresentados (exemplo: descrição de um produto, trecho de plano de projeto, arquivos de entrevista, etc.), e por último é requisitado que o jogador selecione a solução mais adequada dentre a lista de alternativas apresentadas. Baseado na alternativa selecionada, então uma pontuação e um feedback pré-definidos são apresentados. Independente da alternativa que foi selecionada pelo aluno, o jogo avança para o próximo passo, usando sempre a alternativa correta para continuar o jogo. No final, uma pontuação total é calculada baseada na soma das pontuações parciais e um relatório final com todas as pontuações parciais e feedback é gerado.
CONTEXTO DE USO	O público-alvo do jogo são pós-graduandos em cursos de ciência da computação ou profissionais de engenharia de software. O jogo foi projetado para ser usado como um complemento aos cursos tradicionais de sala de aula ou cursos de ensino a distância, proporcionando um ambiente para exercitar os conceitos apresentados. Exige um conhecimento básico de engenharia de software, medição de software, gerenciamento de projetos e do modelo CMMI.
RESULTADO DE APRENDIZAGEM	Conhecimento e habilidade.
EFEITO DE APRENDIZAGEM	Baseado no resultado do teste de hipótese realizado não se pode concluir que o jogo contribui para um efeito de aprendizado positivo, apesar de os alunos que o jogaram terem acreditado que o jogo os ajudou a aprender tanto sobre conceitos e processo quanto sobre a aplicação de medição. Em geral, os participantes avaliaram o jogo como apropriado e atrativo, e muitos participantes comentaram que preferiram o jogo em relação a exercícios tradicionais.
PRINCIPAIS DESCOBERTAS	Para avaliar a efetividade do jogo foi realizada uma série de experimentos dentro de um módulo de medição de software como parte das aulas expositivas de Engenharia de Software em dois cursos de mestrado em Ciência da Computação. Em cada experimento, os alunos foram igualmente treinados em relação aos conceitos básicos de medição e do processo de medição por meio de uma aula expositiva e exercícios em sala de aula pelo mesmo instrutor. Após o treinamento ambos os grupos (experimental e controle) em cada experimento responderam a um pré-teste antes da aplicação do tratamento. O tratamento do uso do jogo só foi aplicado ao grupo experimental, nenhum tratamento foi aplicado a grupo de controle. Após o tratamento ambos os grupos em cada experimento responderam a um pós-teste. No total 15 alunos participaram dos experimentos, então devido ao número pequeno da amostra o teste estatístico utilizado para avaliar a efetividade do jogo foi o teste não paramétrico de Mann-Whitney. Além disso, os participantes dos grupos experimentais responderam a um questionário qualitativo com questões

	abertas sobre o jogo.
ID	12
NOME	U-TEST
REFERENCIAS	SILVA, Antonio Carlos. Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade. 2010. 195 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Itajaí, São José, 2010.
DESCRIÇÃO	Jogo de simulação para apoio ao ensino de teste de software, com foco específico em teste de unidade, abordando questões teóricas e práticas. O jogo permite ao jogador assumir o papel de um testador responsável por escrever teste de unidade para funções já escritas de um sistema hipotético. O objetivo geral do jogo resume-se a aplicação de técnicas para seleção de dados de entrada para o teste de unidade. Durante o jogo o jogador terá de aplicar técnicas de teste de software, como particionamento em classes de equivalência e análise de valor limite. Como resultado ao seu desempenho o jogador será informado de sua colocação no ranking geral de jogadores.
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador na internet.
ÁREA DE CONHECIMENTO	Teste de software.
RESULTADOS ESPERADOS	VER 2 e VAL 2.
ATIVIDADE DE APRENDIZAGEM	Inicialmente o jogador deve selecionar um projeto para “trabalhar” dentre a lista de projetos disponíveis na página de classificados do jornal U-TEST. Ao selecionar um projeto o jogador deve responder uma série de perguntas sobre teste de software simulando uma entrevista de emprego. A cada pergunta é apresentado um feedback ao jogador, seja positivo (quando ele acerta), seja negativo (quando ele erra), e o jogador deve acertar o número mínimo de perguntas para ser contratado. Ao ser contratado é apresentado um artefato para o jogador, que servirá de base para a sequência de desafios que irão explorar conceitos como, por exemplo, classes de equivalência e análise do valor limite. Ao iniciar o seu trabalho os desafios são então apresentados ao jogador como, por exemplo, no desafio 1 o jogador deve montar as partições de equivalência para uma entrada da função definida pelo artefato, e consiste em selecionar dentre as opções disponíveis quais delas correspondem as partições de equivalência desta entrada. Depois de passar pelo desafio, o jogador deverá ainda responder a uma pergunta. Esta pergunta tem como objetivo verificar se o jogador realmente sabe o porquê da sua resposta ao desafio ou se ele simplesmente está selecionado opções aleatórias. No caso de não conseguir responder corretamente a pergunta, ele perde pontuação. Após passar por todos os desafios o jogo termina com apresentação do hall da fama com pontuação de todos os jogadores.
CONTEXTO DE USO	Jogo para ser utilizado por alunos dos cursos da área de Computação/Informática que se encontram cursando a disciplina de Engenharia de Software ou especificamente Teste de Software, e que possuam pelo menos algum embasamento teórico e prático nos principais conceitos envolvidos no teste de software.
RESULTADO DE APRENDIZAGEM	Conhecimento e habilidade.
EFEITO DE APRENDIZAGEM	No experimento 1, a avaliação quantitativa apontou que os alunos que jogaram o jogo U-TEST apresentaram um efeito de aprendizagem superior aos alunos que jogaram o jogo placebo. E a avaliação qualitativa apontou que os alunos gostaram

	<p>da atividade, apesar de apontar certos problemas ou dificuldades, como na contextualização ou dificuldade do jogo.</p> <p>No experimento 2, na avaliação quantitativa não foi possível provar estatisticamente que o jogo U-TEST apresenta um efeito de aprendizagem superior ao exercício aplicado. E a avaliação qualitativa apontou que os alunos gostaram do jogo.</p> <p>No experimento 3, a avaliação quantitativa apontou que apenas no nível de lembrança houve uma melhora no efeito de aprendizagem médio, para os demais níveis observou-se o oposto, resultados do pós-teste inferiores ao do pré-teste. E a avaliação qualitativa também apontou que os alunos gostaram do jogo.</p>
PRINCIPAIS DESCOBERTAS	Para avaliar o efeito de aprendizagem do jogo foram realizados três experimentos, dois deles experimentais, em que os participantes foram distribuídos de maneira aleatória em dois grupos (experimental e controle), e um não-experimental, em que os participantes não foram divididos em grupos. O design do experimento realizado consistiu na aplicação de um pré-teste e um pós-teste, aplicados respectivamente antes e depois de uma seção de tratamento. No primeiro experimento foi realizado um comparativo entre o jogo U-TEST e outro jogo sem relação com o conteúdo da disciplina, como forma de placebo, e participaram deste experimento dez alunos da turma de Engenharia de Software do sexto semestre de um curso de Ciência da Computação. O segundo experimento foi realizado com treze alunos do terceiro semestre de um curso de Tecnólogo em Informática, tendo como principal diferença em relação ao primeiro a aplicação de um exercício prático junto ao grupo de controle abordando o mesmo conteúdo disponibilizado no jogo. O terceiro experimento foi realizado com sete alunos também da turma de Engenharia de Software do sexto semestre de um curso de Ciência da Computação, sendo que neste experimento não houve divisão de grupos e todos os alunos jogaram o jogo U-TEST. Para realizar a avaliação quantitativa foi utilizado teste não-paramétrico de Mann-Whitney. Para realizar a avaliação quantitativa os alunos que jogaram o jogo preencheram um questionário de percepção do jogo.
ID	13
NOME	A Ilha dos Requisitos
REFERENCIAS	GONÇALVES, Rafael Queiroz. Desenvolvimento de um jogo para ensino de engenharia de requisitos. 2010. 148 f. Monografia (Graduação) - Curso de Ciência da Computação, Departamento de Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, São José, 2010.
DESCRIÇÃO	O jogo é contextualizado em um ambiente lúdico onde um analista de requisitos sofre um acidente de avião caindo em uma ilha deserta que possui um vulcão que está prestes a explodir. Esta ilha também é habitada por canibais que acham que o analista de requisitos é um enviado de seu Deus para salvá-los da grande explosão do vulcão que está prestes a ocorrer. Para salvar a si próprio e todos os habitantes da ilha desta explosão, o analista de requisitos deve então conseguir resolver uma série de desafios relacionados com o processo de Engenharia de Requisitos.
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador na internet.
ÁREA DE CONHECIMENTO	Requisitos de Software
RESULTADOS ESPERADOS	GRE 1, GRE 2, GRE 3, GRE 4, GRE 5, DRE 1, DRE 2, DRE 3, DRE 4, DRE 6 e DRE 8.
ATIVIDADE DE	Durante o jogo o jogador de resolver desafios que visam proporcionar ao jogador

APRENDIZAGEM	uma passagem completa por todas as fases do processo de Engenharia de Requisitos. No primeiro desafio o jogador recebe uma série de pedras desorganizadas, cada uma delas representa uma fase do processo de Engenharia de Requisitos, e o jogador deve colocá-las na sequencia correta. No segundo desafio são apresentadas várias habilidades e o jogador deve selecionar somente aquelas ligadas diretamente a um analista de requisitos. No terceiro desafio uma lista de frases é apresentada e o jogador deve selecionar as frases que correspondem ao problema. No quarto desafio uma lista de requisitos funcionais e não-funcionais são apresentados e o jogador deve separar os requisitos funcionais dos não-funcionais arrastando-os para o local reservado para cada tipo de requisito. No quinto desafio o jogador deve lembrar de validar os requisitos indo até o local onde se encontra o fornecedor de requisitos, ao chegar lá o jogador ainda deve selecionar dentre a lista de opções apresentadas, os motivos pelo qual ele foi até o local onde se encontrava o fornecedor de requisitos, e depois responder por que ele considera a validação de requisitos importante. No sexto desafio o jogador recebe uma série de pedras desorganizadas, que correspondem aos passos necessários para gerenciar uma mudança de requisitos adequadamente, e o jogador deve colocá-las na ordem correta. No sétimo e último desafio o jogador deve selecionar dentre a lista de opções apresentadas, o que ele deve fazer depois que uma mudança de requisitos foi aprovada. O jogador vence o jogo se ele conseguir completar todos os desafios no tempo determinado pelo jogo.
CONTEXTO DE USO	Jogo para ser utilizado por alunos de cursos de Computação/Informática para reforçar os conceitos vistos em sala de aula referente a processo de Engenharia de Requisitos.
RESULTADO DE APRENDIZAGEM	Conhecimento.
EFEITO DE APRENDIZAGEM	O resultado dos testes estatísticos comprovou com 95% de certeza que o jogo “A Ilha dos Requisitos” tem eficácia para ensinar Engenharia de Requisitos nos níveis de entendimento e compreensão. O resultado da avaliação qualitativa mostrou que a maioria dos alunos gostou de jogar o jogo “A Ilha dos Requisitos”, e o consideraram relevante para o aprendizado.
PRINCIPAIS DESCOBERTAS	A avaliação do jogo foi feita por meio de um experimento que consistia na divisão de alunos em dois grupos, sendo um de controle e outro experimental, sendo que os dois grupos realizaram um pré-teste e um pós-teste. Entre os dois testes os alunos do grupo experimental jogaram o jogo “A Ilha dos Requisitos”, enquanto os alunos do grupo de controle jogaram outro jogo, sem relação com o conteúdo de Engenharia de Requisitos, como forma de placebo. Participaram do experimento 31 alunos das turmas do segundo e terceiro semestre de um curso técnico em informática. Com base nos resultados dos testes, foi realizada a avaliação do efeito de aprendizagem relativa, baseada na diferença entre o pós-teste e o pré-teste, e também a avaliação do efeito de aprendizagem absoluto, qual considera a quantidade de acertos realizados no pós-teste. Em ambas as avaliações foram utilizadas o teste estatístico não paramétrico de Mann-Whitney para verificar as hipóteses de pesquisa. O teste qualitativo foi obtido através dos questionários respondidos pelos alunos do grupo experimental.
ID	14
NOME	123SPI
REFERENCIAS	ADORNO, Kristiane Silva Vasconcelos de Pina. Jogo educacional para aplicação dos processos do CMMI nível 2 e 3. 2010. 132 f. Qualificação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Itajaí, 2010, 2010.
DESCRIÇÃO	O 123SPI é um jogo de tabuleiro baseado no jogo BELTS Challenge sobre Seis

	Sigma, que foi adaptado para abordar os 18 processos dos níveis 2 e 3 do modelo CMMI. Durante o jogo o jogador deve implantar processos em uma organização alinhado aos níveis 2 e 3 de maturidade do modelo CMMI, e para conseguir implantar esses processos o jogador deve responder corretamente as perguntas feitas sobre os processos presentes no tabuleiro. Vence o jogo quem conseguir implantar primeiro todos os processos definidos em sua carta de missão.
FORMA DE UTILIZAÇÃO	Múltiplos usuários.
TIPO DO TRABALHO	Tabuleiro.
ÁREA DE CONHECIMENTO	Requisitos de Software, Gerência de Projetos, Medição de Software, Qualidade de Software, Gerência de Configuração, Projeto de Software, Teste de Software, Processo de Software, Construção de Software e Manutenção de Software.
RESULTADOS ESPERADOS	Todos os resultados até o nível C.
ATIVIDADE DE APRENDIZAGEM	No 123SPI, cada jogador começa a disputa recebendo uma carta contendo 3 objetivos. Esses objetivos são ligados à melhoria de processos alinhado aos níveis 2 e 3 de maturidade do CMMI e que deverão ser implantados na organização. Cada objetivo indica o número de recursos que devem ser incorporados aos processos para que o objetivo seja cumprido. Para atingir os objetivos e conseguir implantar as melhorias, cada jogador começa o jogo com alguns recursos previamente definidos pela banca que serão ao longo da partida investidos nos processos. Em cada rodada, os jogadores devem mover seus pinos para um processo próximo, que seja ligado pela trilha e, depois que todos se movimentaram a um novo processo, é retirada uma carta de Pergunta. Através dessa carta, um dos participantes ou um moderador lê a carta. Feita a pergunta ao grupo de jogadores, eles devem escolher a carta com a alternativa que consideram corretas, cartas A, B, C ou D. Escolhida a resposta, os jogadores devem selecionar quantos recursos eles irão apostar naquela pergunta e naquele processo em questão. A aposta deve ser de no mínimo 1 recurso e no máximo 4 recursos por rodada. Os jogadores que acertarem a pergunta podem colocar os recursos apostados no processo em questão colocando os recursos em cima dos círculos do tabuleiro relacionados àquele indicador. Como prêmio para o acerto, os jogadores têm direito a retirar uma carta de Recursos. Nessa carta, aparecerá o quanto de recursos o jogador ganhará do patrocinador da organização para poder reinvestir em novas rodadas. Por outro lado, o jogador que errar a pergunta, perde os recursos apostados, esses recursos não são incorporados ao processo e voltam para a banca ou para os recursos do patrocinador e esse jogador não tem direito a pegar uma carta de Recursos. O jogador que primeiro conseguir completar os seus objetivos vence o jogo.
CONTEXTO DE USO	O público alvo deste jogo é formado por profissionais de Engenharia de Software. O jogo é projetado como um complemento às aulas tradicionais provendo um ambiente para exercitar os conceitos apresentados. É suposto que os alunos tenham conhecimento básico em Engenharia de Software, Gerência de Projetos e sobre CMMI ou MPS.BR. O jogo pode ser realizado com no mínimo 2 e no máximo 4 jogadores. Se houver mais pessoas, elas podem se reunir em equipes, nesse caso cada equipe deve ser considerada um jogador.
RESULTADO DE APRENDIZAGEM	Conhecimento.
EFEITO DE APRENDIZAGEM	Os integrantes da equipe que jogou o jogo 123SPI, obtiveram um avanço significativo em relação ao número de acertos nas questões apresentadas. Para os que apenas participaram como ouvintes não jogando efetivamente o jogo, a média de acertos ficou em 36 questões, ou seja, a equipe acertou cerca de 70% das

	<p>questões, 10% a mais em relação aos que jogaram outro jogo. Já a média de acertos para os profissionais que efetivamente jogaram o jogo, ficou em torno de 39 questões, ou seja, acertaram 75% das questões, 15% a mais dos que não jogaram o jogo sobre o assunto CMMI e 5% a mais dos que participaram como ouvintes.</p> <p>Alguns comentários feitos pelos participantes durante a sessão de simulação são apresentados a seguir:</p> <p>“O jogo é uma forma diferente de aprender os conceitos, pois não temos que decorar, conseguimos aprender praticamente brincando.”</p> <p>“Penso que se levasssem esse modelo de treinamento para outras áreas da engenharia de software, teríamos profissionais mais qualificados em razão dessa ajuda que o jogo traz em aprender os conceitos de cada disciplina de uma maneira diferente.”</p> <p>“As empresas desenvolvedores de software poderiam ser mais atuantes no meio acadêmico e apoiar mais a área da pesquisa educacional.”</p> <p>“Pode comercializar esse jogo, tenho muito interesse na área de melhoria de processos e vi que esse jogo vai me ajudar muito a entender mais sobre o modelo CMMI, quero praticar lá em casa com outros colegas da área.”</p> <p>“O fato do jogo ser de tabuleiro aproxima mais as pessoas e faz com que elas pensem em equipe, talvez se fosse eletrônico não teria tido esse entrosamento tão grande.”</p>
PRINCIPAIS DESCOBERTAS	<p>O procedimento adotado no estudo empírico consistiu nas seguintes etapas: Preenchimento do questionário sobre o participante; Sessão de Treinamento; Sessão de Discussão; Sessão de Treinamento; Sessão de Simulação - Jogo; Captura dos dados qualitativos; Sessão de simulado com uso do questionário; e Captura dos dados quantitativos. Na Sessão de Treinamento, os participantes foram treinados nos processos referente aos níveis 2 e 3 do modelo de maturidade CMMI, no jogo utilizado no estudo e nas ações sobre a aplicação dos processos oferecidas por ele. Na sessão de Simulação – Jogo, os participantes foram divididos em dois grupos, onde um grupo jogou o 123SPI e outro grupo jogou um outro jogo como forma de placebo. Na sessão de discussão foram apresentadas diversas abordagens e cenários comuns da melhoria de processos de software, demonstrando exemplos positivos e negativos sobre a implementação de processos em ambiente de desenvolvimento de software, permitindo aos profissionais reconhecer a importância de ambientes previamente definidos, padronizados e monitorados. Em relação aos participantes do estudo, participaram da equipe que jogou o 123 SPI 4 Analistas de Sistemas, 2 Analistas de Testes, 2 Arquitetos de Software, 4 Desenvolvedores e 1 Coordenador Técnico de Projetos, totalizando 13 jogadores. A equipe que jogou o outro jogo era composta por 2 Analistas de Sistemas, 2 Analistas de Testes, 2 Arquitetos de Software, 4 Desenvolvedores, e 1 Analista de Suporte.</p>
ID	15
NOME	Deliver
REFERENCIAS	GRESSE VON WANGENHEIM, C.; SAVI, R.; BORGATTO, A.; DELIVER! - An Educational Game for Teaching Earned Value Management in Computing Courses. <i>Information and Software Technology</i> , Elsevier, pre-print, 2011.
DESCRIÇÃO	<p>Deliver é um jogo de tabuleiro para ensinar Gerenciamento do Valor Agregado no monitoramento e controle da execução de um projeto de software.</p> <p>O propósito do jogo é reforçar conceitos de Gerenciamento de Valor Agregado e exercitar a sua aplicação. Depois de uma sessão do jogo, espera-se que os estudantes estejam mais aptos a lembrarem os nomes dos indicadores chave e fórmulas, assim como diferenciar e interpretar indicadores. Os estudantes devem</p>

	ser capazes de calcular e interpretar índices de performance para rastrear o progresso do projeto e, calcular e interpretar a estimativa no término.
FORMA DE UTILIZAÇÃO	Múltiplos usuários.
TIPO DO TRABALHO	Tabuleiro.
ÁREA DE CONHECIMENTO	Gerência de Projetos
RESULTADOS ESPERADOS	GPR 4, GPR 5, GPR 6, GPR 7, GPR 11, GPR 13, GPR 14 e GPR 17.
ATIVIDADE DE APRENDIZAGEM	Os participantes iniciam o jogo planejando, através da aquisição de membros para equipe do projeto do reservatório de recursos humanos disponíveis. Recursos humanos podem somente serem adquiridos/demitidos neste momento ou em um marco durante a execução do projeto. Recursos humanos são caracterizados por seu nível de produtividade e seu salário semanal. O nível de produtividade multiplicado pelo valor do dado determina o número de espaços que a equipe do projeto irá avançar no tabuleiro em cada rodada do jogo. A duração do projeto é então estimada pelos jogadores através do número de rodadas necessárias para terminar o projeto. Cada rodada é contada como uma semana no jogo. O custo do projeto é estimado considerando o salário dos recursos humanos escolhidos e a duração estimada do projeto. Os participantes podem também planejar uma reserva para riscos desconhecidos. Os resultados são documentados no plano de projeto, identificando a duração estimada, custo de recursos humanos e reserva. Somando o custo total do projeto, o orçamento para término (BAC) é calculado. Durante a execução do jogo, os pares de jogadores avançam no tabuleiro, simulando a execução do projeto, jogando os dados. Se o número do dado for entre 1 e 4, o valor do dado deve multiplicado com a soma da produtividade da equipe e deve ser avançado o numero correspondente de espaços no tabuleiro. Se o numero do dado for 5 ou 6 uma carta de risco deve ser retirada e as instruções dessa carta deve ser seguida. Em cada marco entre as fases do projeto, o par de jogadores deve monitorar e controlar a execução do projeto usando o EVM. Neste ponto eles devem realizar uma reunião de status analisando a performance do projeto (SPI e CPI) e realizar uma previsão (EAC). Baseado na performance monitorada, eles devem demitir ou adquirir recursos humanos. O vencedor é o par de jogadores que primeiro entregar o projeto para o cliente chegando no espaço de entrega sem ultrapassar o orçamento do projeto.
CONTEXTO DE USO	O jogo é destinado para ser usado como parte de um curso de gerenciamento de projetos. Ele é destinado para ensinar estudantes de programas de computação, incluindo Ciência da Computação, Sistemas da Informação ou cursos de Engenharia de Software. Ele é destinado para ser uma estratégia complementar a uma unidade de instrucional de monitoramento e controle de projeto. Um pré-requisito para a aplicação do jogo é que os estudantes tenham entendimento básico em Gerência de Projetos e Gerenciamento do Valor Agregado ensinados de antemão, por exemplo, através de lesões expositivas.
RESULTADO DE APRENDIZAGEM	Conhecimento, habilidade e atitude.
EFEITO DE APRENDIZAGEM	Baseado na própria avaliação dos estudantes, o jogo ajudou a reforçar e ensinar o cálculo de indicadores de performance, assim como ilustrou e motivou o uso do EVM. Foi observado um resultado de aprendizagem adicional, assim que os estudantes perceberam a baixa precisão das estimativas no início de um projeto, eles melhoraram suas estimativas durante a execução do projeto. No geral, a avaliação do jogo foi bastante positiva com classificação acima de zero pela

	maioria dos estudantes. A avaliação do jogo também confirmou a hipótese que o jogo educacional motiva altamente os estudantes e providencia uma experiência muito positiva. Os estudantes participaram efetivamente da atividade durante o jogo e expressaram que jogar o jogo foi uma experiência agradável. Especialmente a interação social provocada pelo jogo foi comentada como muito positiva. O design do jogo foi considerado como atrativo lidando com conteúdo relevante e conectado com a unidade instrucional e conhecimento prévio adquirido. Os estudantes também demonstraram e confirmaram um alto grau de imersão e diversão durante o jogo. E, apesar do jogo requerer certa quantidade de explicação de ante-mão, a complexidade das regras do jogo foi considerada apropriada e todos os estudantes conseguiram jogar o jogo na primeira tentativa. Ainda, aspectos que receberam classificações baixas e que podem indicar oportunidades de melhorias, incluem a redução da dependência da sorte, assim como, o andamento e a apresentação apropriada dos desafios durante o jogo.
PRINCIPAIS DESCOBERTAS	A avaliação do jogo foi baseada no modelo de Kirkpatrick com foco no nível 1 (reação). O jogo foi avaliado baseado na percepção dos estudantes obtido através de questionários respondidos após a aplicação do jogo. O jogo foi aplicado em um curso de Gerência de Projetos de Software do programa de graduação em Sistemas da Informação e no curso de Gerência de Projetos de Software do programa de graduação de Ciência da Computação no Departamento de Informática e Estatística da Universidade Federal de Santa Catarina durante o segundo semestre de 2010.
ID	16
NOME	Jogo das 7 Falhas
REFERENCIAS	DINIZ, Lúcio Lopes. Jogo das 7 Falhas - Um Jogo Educacional para o Apoio ao Ensino do Teste de Caixa-Preta. 2011. 230 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Itajaí, São José, 2011.
DESCRIÇÃO	O Jogo das 7 Falhas é um jogo educacional de simulação que reproduz a execução de casos de teste derivados a partir de especificações de requisitos, utilizando as técnicas classe de equivalência e análise de valor-limite. O jogo consiste em descobrir as falhas existentes nas funcionalidades de um software a ser testado em menos tempo possível. O jogo possui dois níveis de complexidade: baixa e média. Cada nível é composto por uma funcionalidade onde existem sete falhas a serem descobertas. O jogador só passará para o nível 2 caso descubra as sete falhas existentes no nível 1 dentro do tempo estimado (25 minutos para o nível 1; 40 minutos para o nível 2). Caso o tempo se esgote antes de o jogador identificar as sete falhas em cada nível, o jogo se encerra, e o jogador é eliminado. Caso o jogador descubra as sete falhas do nível 1 e as sete falhas do nível 2 dentro do tempo, o jogador é o vencedor, tendo alcançado o objetivo proposto pelo jogo.
FORMA DE UTILIZAÇÃO	Usuários individuais.
TIPO DO TRABALHO	Computador na internet.
ÁREA DE CONHECIMENTO	Teste de software.
RESULTADOS ESPERADOS	VER 2, VER 4, VER 5, VER 6, VAL 2, VAL 4, VAL 5, VAL 6 e VAL 7.
ATIVIDADE DE APRENDIZAGEM	Ao iniciar o jogo, o jogador terá 25 minutos para identificar as sete falhas existentes na funcionalidade “Cadastro de Usuário”, que estarão em desacordo com os requisitos da funcionalidade. Para identificar as falhas, o usuário irá executar os casos de teste elaborados anteriormente através da utilização das técnicas de classe de equivalência e análise de valor-limite. A cada caso de teste

	executado que não originar uma falha, será exibida, na parte inferior, a mensagem correspondente ao resultado esperado e um feedback informando a qual classe de equivalência ou valor-limite o caso de teste executado corresponde. Para cada caso de teste executado que originar uma falha, será exibido, na parte inferior, uma mensagem informando que uma falha foi descoberta e que é necessário selecionar a classe de equivalência ou o valor-limite que a originou. Também será exibida, abaixo da mensagem, uma lista com sete classes de equivalência e/ou valores-limite, para que o jogador selecione, entre elas, a que deu origem à falha simulada. Caso o jogador selecione a classe de equivalência ou o valor-limite correto, ele receberá dez pontos. Caso o jogador selecione a classe de equivalência ou o valor-limite errado, ele perderá dez pontos. No nível 1, existe ainda a possibilidade de o jogador obter uma dica sobre a localização de uma das falhas, bastando, para isso, pressionar o botão Dicas. Porém, ao pressionar esse botão, o jogador perde 4 pontos. Caso o jogador encontre as sete falhas dentro dos 25 minutos, ele será promovido à analista de teste pleno, e passará para o segundo nível do jogo. Além disso, o jogador ganhará um ponto a cada intervalo de 10 segundos que sobrar do tempo disponível. A mecânica do nível 2 é a mesma do nível 1. A diferença realmente está no grau de complexidade das falhas. Caso descubra as sete falhas do nível dois dentro do intervalo de 40 minutos, será promovido a analista de teste sênior e sairá como vencedor.
CONTEXTO DE USO	O público-alvo do Jogo das 7 Falhas são alunos de graduação em Ciência da Computação e Sistemas de Informação, bem como profissionais de software já formados das áreas de Ciência da Computação e Sistemas de Informação, que estejam ou não trabalhando com testes de software. Embora o jogo seja voltado para pessoas iniciantes em teste de software, é necessário que os jogadores possuam alguns conceitos básicos de testes de software, saibam diferenciar erros, defeitos e falhas, conheçam teste de sistema e a técnica de teste de caixa-preta. Também é pré-requisito que o jogador conheça o que é um caso de teste e as técnicas de seleção de casos de teste: classe de equivalência e análise de valor-limite.
RESULTADO DE APRENDIZAGEM	Conhecimento, habilidade e atitude.
EFEITO DE APRENDIZAGEM	No primeiro experimento não foi possível comprovar a hipótese que o efeito de aprendizagem nos níveis de compreensão, aplicação e análise nos alunos que utilizaram o jogo é superior ao dos alunos que não utilizaram o jogo. Mas foi possível comprovar que o jogo ajuda a desenvolver as habilidades de execução de casos de teste e de encontrar falhas, além de ter sido considerado uma atividade de ensino motivadora. No experimento 2 na análise realizada com as diferenças das notas obtidas entre o pré-teste e o pós-teste, embora mais participantes tenham apresentado um coeficiente de aprendizagem negativo, o estudo resultou em um fator de aprendizagem médio de $\approx 0,00$. Embora a maioria dos participantes tenha achado o jogo atrativo, agradável, com boa duração e com nível de dificuldade adequado, a quantidade foi bem inferior à dos experimentos 1 e 3. Dentre os pontos discutidos sobre o por que desses resultados estão: a falta de atividade de derivação de casos de teste (durante i experimento, alguns participantes reclamaram de não terem tido tempo para ler os requisitos e montar os casos de teste) e problemas apresentados na conexão do servidor no dia de aplicação do experimento. No terceiro experimento foi possível comprovar a hipótese que o efeito de aprendizagem nos níveis de compreensão, aplicação e análise nos alunos que utilizaram o jogo é superior ao dos alunos que não utilizaram o jogo. Também foi possível comprovar que o jogo ajuda a desenvolver as habilidades de execução de casos de teste e de encontrar falhas, além de ter sido considerada uma atividade de ensino motivadora.

PRINCIPAIS DESCOBERTAS	Para avaliação do jogo foram realizados três experimentos, um com 11 alunos da turma de Engenharia de Software 1 do curso de Ciência da Computação, um com 7 alunos da turma de Engenharia de Software 2 do curso de Ciência da Computação, e um com 21 alunos do curso de Ciência da Computação. Nos experimentos 1 e 3 os alunos foram divididos em grupo de controle e grupo experimental, os alunos do grupo experimental jogaram o Jogo das 7 Falhas e os alunos do grupo de controle realizaram exercícios sobre classe de equivalência e análise de valor limite. No experimento 2 não houve divisão de grupos devido ao baixo número de participantes, e todos jogaram o Jogo das 7 Falhas. Em todos os experimentos os alunos responderam a um pré-teste, antes do tratamento, e um pós-teste, após o tratamento. Os alunos que jogaram o jogo responderam também um questionário de avaliação do jogo após a finalização do mesmo. Para análise dos resultados dos experimentos 1 e 3 foi aplicado o teste estatístico de Mann-Whitney. No experimento 2 não foi possível aplicar o teste de Mann-Whitney pois não houve divisão de grupos.
-----------------------------------	---

APÊNDICE B – EVENTOS DO JOGO

Os eventos são situações, relacionadas com os resultados esperados do MPS.BR, que o jogador deverá resolver durante o jogo. Neste apêndice são apresentados os seis eventos disponíveis nesta versão do jogo.

Tabela 30. Evento 1 - definição da forma de trabalho

EV.1. Definição da forma de trabalho.	
Descrição	Seus desenvolvedores estão com uma dúvida sobre como eles devem trabalhar. Minha sugestão é que você vá até a sala de desenvolvimento e converse com eles.
Condição	Depois que o jogador tiver contratado pelo menos um analista e um programador.
Personagem	Desenvolvedores.
Ação	Conversar.
Diálogos	Resultados Esperados
Personagem: EV.1.1. Estamos em dúvida sobre por onde começar?	GPR1 = P
Jogador: EV.1.1.1. Iremos montar um cronograma.	
Feedback: EV.1.1.1.1. Um cronograma é importante, mas para elaborá-lo, você precisa antes saber o que deve ser feito. Pense um pouco mais e converse novamente com sua equipe.	
Feedback: EV.1.1.2. Iremos definir o escopo da próxima versão do produto.	
Feedback: EV.1.1.2.1. Boa idéia. Um projeto deve ter um escopo bem definido para que seja possível identificar aquilo que deve ser entregue e estabelecer um prazo final para sua conclusão.	
EV.1.1.3. Ainda não sei.	
Feedback: EV.1.1.3.1. Pense um pouco mais, mas cuidado para não deixar sua equipe sem saber o que fazer.	
Personagem: EV.1.2. Escopo? Eu já ouvi este termo antes, mas sempre tenho dúvidas sobre o que ele realmente significa.	GPR 1 = P
Jogador: EV.1.2.1. O escopo do projeto pode ser entendido como sendo todos os requisitos que serão desenvolvidos.	

<p>Feedback:</p> <p>EV.1.2.1.1. Cuidado! Quando falamos do escopo do projeto, devemos pensar no trabalho que deve ser realizado para entregar os requisitos do sistema. Revise sua resposta e converse novamente com sua equipe.</p> <p>EV.1.2.2.O escopo do produto pode ser entendido como sendo todas as atividades que precisam ser realizadas no projeto.</p> <p>Feedback:</p> <p>EV.1.2.2.1. Cuidado! Quando falamos do escopo do produto, devemos pensar mais naquilo que deve ser entregue (características e funções do produto). Pense um pouco mais e converse novamente com sua equipe.</p> <p>EV.1.2.3.O escopo é uma descrição detalhada daquilo que iremos fazer e entregar no projeto. Por isso, temos escopo de projeto e escopo de produto.</p> <p>Feedback:</p> <p>EV.1.2.3.1. Perfeito! Quando falamos do escopo do projeto, pensamos no trabalho que deve ser realizado para entregar os requisitos do sistema. No escopo do produto, devemos pensar naquilo que deve ser entregue (características e funções do produto). Assim, precisamos saber o que iremos fazer (produto) e como iremos fazer (projeto).</p>	
<p>Personagem:</p> <p>EV.1.3. Entendi, mas você poderia nos mostrar exemplos de escopo de produto?</p> <p>Jogador:</p> <p>EV.1.3.1.Estrutura analítica de projeto (EAP – também conhecida pelo termo em inglês WBS – Work Breakdown Structure), listagem das tarefas do projeto, coluna das tarefas nas diferentes sprints de um projeto ágil, ou seja, qualquer documento que identifique o trabalho a ser realizado para garantir o escopo do produto.</p> <p>Feedback:</p> <p>EV.1.3.1.1. Cuidado! Quando falamos do escopo do produto, devemos pensar mais naquilo que deve ser entregue (características e funções do produto). Pense um pouco mais e converse novamente com sua equipe.</p> <p>EV.1.3.2.Especificação de requisitos do sistema, product backlog, qualquer documento que especifique as características e funcionalidades do produto.</p> <p>Feedback:</p> <p>EV.1.3.2.1. Excelente! Você realmente entende o que é um escopo de produto.</p>	GPR1 = L
<p>Personagem:</p> <p>EV.1.4. E sobre escopo de projeto, você também teria exemplos?</p> <p>Jogador:</p> <p>EV.1.4.1.Documento de requisitos do projeto ou qualquer documento que especifique as características e funcionalidades do projeto.</p> <p>Feedback:</p> <p>EV.1.4.1.1. Cuidado! Quando falamos do escopo do projeto, devemos pensar no trabalho que deve ser realizado para entregar os requisitos do sistema. Revise sua resposta e converse novamente com sua equipe.</p>	GPR1 = T

<p>EV.1.4.2.Estrutura analítica de projeto (EAP – também conhecida pelo termo em inglês WBS – Work Breakdown Structure), listagem das tarefas do projeto, coluna das tarefas nas diferentes sprints de um projeto ágil, ou seja, qualquer documento que identifique o trabalho a ser realizado para garantir o escopo do produto.</p> <p>Feedback:</p> <p>EV.1.4.2.1. Isso mesmo! Em pouco tempo, você nem precisará mais de mim.</p>	
<p>Personagem:</p> <p>EV.1.5. Então iremos começar pela definição dos requisitos do sistema, certo?</p> <p>Jogador:</p> <p>EV.1.5.1.Isto mesmo. Já vou definir quem será o analista e pedir a ele que levante os requisitos com nosso fornecedor de requisitos.</p> <p>Feedback:</p> <p>EV.1.5.1.1. Tome mais cuidado. Concordo que levantar os requisitos é fundamental, mas você deve pensar primeiro em como o trabalho será desenvolvido. Como será o levantamento dos requisitos, quem serão os envolvidos, quais artefatos serão gerados, etc. Reveja sua forma de pensar e volte a conversar com sua equipe.</p> <p>EV.1.5.2.Não exatamente. Antes de iniciarmos o levantamento dos requisitos, nós precisamos definir qual será o ciclo de vida que adotaremos para o projeto.</p> <p>Feedback:</p> <p>EV.1.5.2.1. Boa idéia. Assim, você poderá identificar como as atividades do projeto serão organizadas.</p>	GPR3 = P
<p>Personagem:</p> <p>EV.1.6. Mas por que precisamos de um ciclo de vida?</p> <p>Jogador:</p> <p>EV.1.6.1.A empresa será avaliada futuramente no MPS.BR. Ter um ciclo de vida definido é uma das exigências do modelo.</p> <p>Feedback:</p> <p>EV.1.6.1.1. Quase fiquei com vontade de te dar um corretivo! Mas, vamos lá. Você não deve pensar em melhoria para atender a um modelo, mas para efetivamente melhorar sua forma de trabalho.</p> <p>EV.1.6.2.Com o ciclo de vida, podemos pensar em fases, atividades, responsáveis e produtos de trabalho (artefatos) que iremos produzir ao longo do projeto.</p> <p>Feedback:</p> <p>EV.1.6.2.1. Ótimo argumento. Para realizar um projeto com sucesso, você precisa saber o que deve ser feito, como e quando, e ainda, por quem será feito. Um ciclo de vida adequado apoiará também a elaboração do escopo do projeto (as tarefas virão do ciclo de vida).</p>	GPR3 = P
<p>Personagem:</p> <p>EV.1.7. Você não acha que vale a pena já iniciarmos o levantamento dos requisitos enquanto estamos definindo o ciclo de vida.</p> <p>Jogador:</p>	GPR7 = P GPR8 = P GPR9 = P

<p>EV.1.7.1.Ok, seu argumento faz sentido. Concordo que podemos dividir os esforços e trabalhar em paralelo.</p> <p>Feedback:</p> <p>EV.1.7.1.1. Isso pode agilizar o trabalho, mas pense bem, como iremos iniciar o levantamento dos requisitos sem saber como iremos fazê-lo, quais os envolvidos, ferramentas que serão adotadas, templates, etc? Revise sua resposta e volte a conversar com o gerente de projetos.</p> <p>EV.1.7.2.Entendo sua preocupação, mas como iremos iniciar o levantamento dos requisitos sem saber como iremos fazê-lo, quais os envolvidos, ferramentas que serão adotadas, templates, etc. Precisamos planejar todo o trabalho a ser feito.</p> <p>Feedback:</p> <p>EV.1.7.2.1. Parabéns você realmente entende que o trabalho precisa ser planejado antes de ser realizado.</p>	
<p>Personagem:</p> <p>EV.1.8. Obrigado pelas explicações chefe. Ficaremos no aguardo destas definições.</p> <p>Jogador:</p> <p>EV.1.8.1.De nada.</p>	

Tabela 31. Evento 2 - definição do ciclo de vida

EV.2. Definição do ciclo de vida.	
Descrição	O gerente de projetos quer conversar com você, ele está com dúvida sobre qual modelo de ciclo de vida utilizar.
Condição	Um dia após o evento 1.
Personagem	Gerente de projetos.
Ação	Conversar.
Diálogos	Resultados Esperados
Personagem:	GPR3 = P
EV.2.1. Oi chefe, você chegou a pensar na duração dos nossos projetos de evolução do produto?	
Jogador:	
EV.2.1.1.Sim, pensei. Acredito que devemos nos preparar para lançar 4 releases anuais. Assim, os projetos teriam uma duração média de 3 meses.	
Personagem:	GPR 3 = L
EV.2.2. Interessante. Acredito que isso organizaria melhor nosso trabalho. O que você acha então de adotarmos o modelo de ciclo de vida cascata em nossos projetos?	
Jogador:	
EV.2.2.1.O modelo cascata é uma boa sugestão. Teremos flexibilidade para alterar os requisitos ao longo do desenvolvimento de cada release.	
Feedback:	

<p>EV.2.2.1.1. Tome cuidado! Se você quer flexibilidade para alterar os requisitos durante os projetos, então o modelo cascata não é uma boa escolha. Pois no modelo cascata os requisitos são definidos no início do projeto e não devem ser alterados até o final do mesmo. Revise sua resposta e volte a conversar com o gerente de projetos.</p> <p>EV.2.2.2.O modelo cascata poderia ser uma boa se nossos requisitos fossem estáveis. Porém, acredito que precisamos ter mais flexibilidade para comportar mudanças numa release.</p> <p>Feedback:</p> <p>EV.2.2.2.1. Isso mesmo! O modelo cascata é ideal para projetos onde os requisitos não mudam. Se seus requisitos não são estáveis, você deve pensar em um modelo de ciclo que permita ter mais flexibilidade para comportar mudanças.</p>	
<p>Personagem:</p> <p>EV.2.3. Entendi! Então o que você acha de utilizarmos o modelo iterativo?</p> <p>Jogador:</p> <p>EV.2.3.1.O modelo iterativo se encaixa melhor, pois podemos ter entregas intermediárias em cada release que nos permitirão gerenciar melhor mudanças e as expectativas do nosso gerente de produto.</p> <p>Feedback:</p> <p>EV.2.3.1.1. Muito bem! Realmente o modelo iterativo se encaixa melhor com sua empresa, pois como você tem requisitos instáveis você poderá fazer mudanças nos mesmos a cada iteração.</p> <p>EV.2.3.2.O modelo iterativo não é uma boa escolha, pois se fizermos entregas intermediárias a cada iteração, o gerente de produto irá pedir muitas modificações nas releases.</p> <p>Feedback:</p> <p>EV.2.3.2.1. Veja bem, não é por que o ciclo de vida iterativo é flexível a mudanças que você deve sair fazendo mudanças de qualquer jeito. Essas mudanças precisam ser controladas, ou seja, você deve avaliar o impacto das mudanças para que se possa decidir junto ao gerente de produto se a mudança deve ser realizada ou não. Pense um pouco mais e volte a conversar com o gerente de projetos.</p>	GPR3 = L
<p>Personagem:</p> <p>EV.2.4. O que você quer dizer com entregas intermediárias? Ao final de cada iteração temos que entregar um release do produto? Mas e a idéia de 4 releases por ano?</p> <p>Jogador:</p> <p>EV.2.4.1.As entregas intermediárias podem ser entregas internas que nos permitirão avaliar com o gerente de produto se estamos desenvolvendo o produto certo.</p> <p>Feedback:</p> <p>EV.2.4.1.1. Boa! Quando se trabalha com entregas intermediárias a equipe fica mais segura, pois ela consegue ter uma melhor percepção sobre o desenvolvimento do produto. Além disso, as entregas intermediárias permitem</p>	GPR3 = L

<p>que falhas sejam identificadas e corrigidas antes do final do projeto.</p> <p>EV.2.4.2. Você tem razão. Acho que seria melhor adotarmos sua sugestão inicial e trabalhar com o modelo cascata.</p> <p>Feedback:</p> <p>EV.2.4.2.1. Cuidado! Você não precisa entregar uma release a cada final de iteração, você pode ter entregas internas onde a própria equipe junto com o gerente de produto irão avaliar o produto. Esta avaliação permite que problemas sejam identificados e corrigidos antes do final do projeto. Pense nisso e volte a conversar com o gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.2.5. Certo, então agora que já definimos o ciclo de vida que iremos utilizar, podemos começar a desenvolver nosso produto?</p> <p>Jogador:</p> <p>EV.2.5.1. Podemos sim, só não se esqueça de fazer uma apresentação do nosso ciclo de vida para a equipe.</p> <p>Feedback:</p> <p>EV.2.5.1.1. Tenha calma, você está esquecendo das fases do ciclo de vida. A definição das fases permite organizar melhor o projeto e fornecem uma base para a definição das atividades do mesmo, além de permitir períodos planejados de avaliação e de tomada de decisões. Pense mais um pouco e volte a conversar com o gerente de projetos.</p> <p>EV.2.5.2. Ainda não, você já pensou nas fases que irão compor o nosso ciclo de vida para o projeto de cada release?</p> <p>Feedback:</p> <p>EV.2.5.2.1. Bem lembrado! A definição das fases do projeto permite organizar melhor o projeto e fornecem uma base para a definição das atividades do mesmo, além de permitir períodos planejados de avaliação e de tomada de decisões.</p>	GPR3 = L
<p>Personagem:</p> <p>EV.2.6. Fases do projeto? Ainda não tinha pensado nisso, você poderia me dar exemplos destas fases?</p> <p>Jogador:</p> <p>EV.2.6.1. Iniciação, planejamento, construção e encerramento.</p> <p>Feedback:</p> <p>EV.2.6.1.1. Perfeito, essas fases realmente correspondem a fases de um ciclo de vida de projeto. Lembre-se que o ciclo de vida do projeto corresponde ao que você precisa fazer para disponibilizar um produto, serviço ou entrega. Já o ciclo de vida de produto, corresponde toda a vida útil do produto, desde o início da elaboração até o seu descarte. Observe que podemos ter diversos projetos durante o ciclo de vida de um produto.</p> <p>EV.2.6.2. Concepção, crescimento, maturidade, declínio e retirada.</p> <p>Feedback:</p>	GPR3 = L

<p>EV.2.6.2.1. Preste mais atenção! Essas fases são exemplos de fases de ciclo de vida de produto e não de projeto. Lembre-se que o ciclo de vida do projeto corresponde o que você precisa fazer para disponibilizar um produto, serviço ou entrega. E o ciclo de vida de produto, corresponde toda a vida útil do produto, desde o início da elaboração até o seu descarte. Observe que podemos ter diversos projetos durante o ciclo de vida de um produto. Revise sua resposta e volte a conversar com gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.2.7. Entendi. Precisaremos definir também atividades para cada uma destas fases, certo?</p> <p>Jogador:</p> <p>EV.2.7.1. Isso. Lembra do escopo de projeto? Agora iremos descrever as atividades de cada fase, pensando também na sequência e dependência destas atividades.</p> <p>Feedback:</p> <p>EV.2.7.1.1. Isso mesmo! Lembre-se que o escopo do projeto é todo o trabalho necessário para concluir o projeto com sucesso. Por isso você deve definir todas as atividades necessárias assim como as dependências entre elas para que depois possamos elaborar o cronograma.</p> <p>EV.2.7.2. Isso mesmo. Assim, teremos o escopo do produto bem definido. Devemos pensar também na sequência e dependência destas atividades.</p> <p>Feedback:</p> <p>EV.2.7.2.1. Como você está desatento! Você deve lembrar que quando falamos em atividades estamos falando de escopo de projeto e não de produto, já havíamos conversado sobre isso antes. Pense mais um pouco e volte a conversar com o gerente de projetos.</p>	GPR3 = T
<p>Personagem:</p> <p>EV.2.8. Mas como iremos saber quem pode fazer qual atividade?</p> <p>Jogador:</p> <p>EV.2.8.1. Boa pergunta. Para cada atividade, iremos também definir o papel responsável por ela, além dos conhecimentos e habilidades esperados para o papel.</p> <p>Feedback:</p> <p>EV.2.8.1.1. Perfeito! Assim, você saberá quais os conhecimentos e habilidades necessários para executar cada atividade. Mas lembre-se que você ainda precisa de algum documento que guarde o currículo de cada colaborador e que indique quais papéis cada colaborador está habilitado a executar. Assim você saberá quais pessoas poderão executar cada atividade.</p> <p>EV.2.8.2. Iremos atribuir um responsável por cada atividade com base no nosso conhecimento sobre a equipe. Nossa equipe é pequena, nós sabemos quais são os conhecimentos e habilidades de cada membro da equipe.</p> <p>Feedback:</p> <p>EV.2.8.2.1. Tome cuidado! É importante que você defina os conhecimentos e habilidades necessários para executar cada atividade, para que no futuro você saiba por que uma determinada pessoa foi alocada para uma atividade, ou se</p>	GPR7 = P

<p>for preciso substituir a pessoa que está executando uma atividade você saiba quais habilidades e conhecimentos a nova pessoa deve possuir. Revise sua resposta e volte a conversar com o gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.2.9. E se houverem outros envolvidos nas atividades?</p> <p>Jogador:</p> <p>EV.2.9.1. Vamos simplificar e ter apenas um papel para cada atividade.</p> <p>Feedback:</p> <p>EV.2.9.1.1. Não mesmo! Neste caso você precisa determinar todos os papéis envolvidos com a atividade e determinar o papel responsável. Por exemplo, na atividade de validação de requisitos podem participar o analista de requisitos e o gerente de produto, e o papel responsável será o analista de requisitos. Pense mais um pouco e volte a conversar com o gerente de projetos.</p> <p>EV.2.9.2. Bem lembrado. Neste caso, devemos ter um papel responsável pela atividade, mas devemos descrever também todos os demais papéis. Quando detalharmos a atividade, devemos explicar a participação de cada papel.</p> <p>Feedback:</p> <p>EV.2.9.2.1. Isso mesmo! Nos casos que você tem mais pessoas envolvidas em uma atividade, você deve determinar todos os papéis que estarão envolvidos com a atividade e determinar o papel responsável. Por exemplo, na atividade de validação de requisitos podem participar o analista de requisitos e o gerente de produto, e o papel responsável será o analista de requisitos.</p>	GPR7 = P
<p>Personagem:</p> <p>EV.2.10. E o que eu devo colocar como conhecimentos?</p> <p>Jogador:</p> <p>EV.2.10.1. Você deve colocar como conhecimentos tudo que um colaborador deve conhecer para poder executar uma função. Por exemplo, para o papel programador Java o colaborador deve conhecer programação orientada a objetos, lógica de programação, sintaxe do Java, UML, etc.</p> <p>Feedback:</p> <p>EV.2.10.1.1. Isso mesmo, o conhecimento está relacionado com o que o colaborador conhece e não com o que ele sabe fazer. Por exemplo, um colaborador pode ter conhecimentos sobre uma linguagem de programação, mas não sabe programar, por que nunca colocou em prática o que ele aprendeu.</p> <p>EV.2.10.2. O conhecimento está relacionado com o que um colaborador deve saber fazer para poder executar um determinado papel, por exemplo, para o papel programador Java o colaborador deve saber programar em Java, deve saber ler e interpretar diagramas UML, deve saber trabalhar em equipe, etc.</p> <p>Feedback:</p> <p>EV.2.10.2.1. Tome cuidado! O conhecimento está relacionado com o que um colaborador conhece e não com o que ele sabe fazer. Por exemplo, um colaborador pode ter conhecimentos sobre uma linguagem de programação, mas não sabe programar, por que nunca colocou em prática o que ele aprendeu.</p>	GPR7 = L

<p>Reflita sobre isso e volte a conversar com o gerente de projetos.</p> <p>Personagem:</p> <p>EV.2.11. E como habilidades, o que eu devo colocar?</p> <p>Jogador:</p> <p>EV.2.11.1. A habilidade está relacionada com as características pessoais de um colaborador, como por exemplo, agilidade, criatividade, comprometimento, entusiasmo, ética, etc.</p> <p>Feedback:</p> <p>EV.2.11.1.1. Você está errado. A habilidade está relacionada com o que um colaborador sabe fazer, e não com suas características pessoais. Pense nisso e volte a conversar com o gerente de projetos.</p> <p>EV.2.11.2. Você deve colocar como habilidade tudo que um colaborador deve saber fazer para executar uma função. Utilizando o exemplo anterior, para o papel programador Java, é necessário que o colaborador saiba programar utilizando a linguagem de programação Java, possua boa comunicação, saiba trabalhar em equipe, saiba interpretar diagramas UML, etc.</p> <p>Feedback:</p> <p>EV.2.11.2.1. Isso mesmo! A habilidade está relacionada com o que um colaborador sabe fazer, e não com suas características pessoais.</p>	GPR7 = L
<p>Personagem:</p> <p>EV.2.12. Entendi. Assim, ficará mais fácil verificar qual colaborador da equipe pode executar uma determinada atividade. Mas, e se não tivermos alguém com os conhecimentos e habilidades para executar um papel no projeto?</p> <p>Jogador:</p> <p>EV.2.12.1. Neste caso, teremos que contratar alguém com o perfil desejado.</p> <p>Feedback:</p> <p>EV.2.12.1.1. Tome cuidado, você não é obrigado a contratar só por que nenhum membro da sua equipe possui o perfil desejado para executar um determinado papel, você pode treinar um colaborador ao invés de contratar. Veja bem, eu não estou dizendo que você não possa contratar só que não é obrigatório. Pense um pouco mais e volte a conversar com o gerente de projetos.</p> <p>EV.2.12.2. Neste caso, podemos contratar alguém com o perfil desejado ou podemos planejar um treinamento adicional.</p> <p>Feedback:</p> <p>EV.2.12.2.1. É isso aí! Você deve avaliar o que é melhor para sua empresa para decidir se você vai treinar ou contratar alguém.</p> <p>EV.2.12.3. Neste caso, teremos que treinar alguém para atender ao perfil desejado.</p> <p>Feedback:</p> <p>EV.2.12.3.1. Tome cuidado, você não é obrigado a treinar alguém quando nenhum membro da sua equipe possua o perfil desejado para executar um</p>	GPR7 = T

<p>determinado papel, você pode contratar ao invés de treinar. Veja bem, eu não estou dizendo que você não possa treinar só que não é obrigatório. Pense um pouco mais e volte a conversar com o gerente de projetos.</p> <p>Personagem:</p> <p>EV.2.13. Legal chefe. Você manja mesmo deste negócio de melhoria de processo! Obrigado pela ajuda.</p> <p>Jogador:</p> <p>EV.2.13.1. De nada.</p>	
---	--

Tabela 32. Evento 3 - definição do que colocar no plano de projeto

EV.3. Definição do que colocar no plano de projeto.	
Descrição	O gerente de projetos está iniciando o planejamento do primeiro projeto e está com dúvida sobre o que colocar no plano de projeto. Vá até a sala de desenvolvimento e converse com ele.
Condição	Um dia após o evento 2.
Personagem	Gerente de projetos.
Ação	Conversar.
Diálogos	Resultados Esperados
Personagem:	
EV.3.1. Oi Chefe! Agora que já temos o escopo de nosso primeiro projeto definido, eu já posso começar a planejá-lo, certo?	
Jogador:	
EV.3.1.1. Pode sim.	
Personagem:	GPR 2 = P
EV.3.2. Certo! Por onde eu devo começar?	
Jogador:	
EV.3.2.1. Podemos partir para a elaboração do cronograma.	
Feedback:	
EV.3.2.1.1. A elaboração do cronograma é importante, mas como você vai elaborar o cronograma se você ainda não sabe todas as tarefas do projeto e nem o tamanho destas tarefas? Pense mais um pouco e volte a conversar com o gerente de projeto.	
EV.3.2.2. Vamos estimar o tamanho do projeto primeiro.	
Feedback:	
EV.3.2.2.1. Bem pensado! A estimativa de tamanho é importante, pois no futuro podemos calcular as demais estimativas do projeto a partir da estimativa de tamanho.	
Personagem:	GPR2 = L

<p>EV.3.3. Certo, e como vamos estimar o tamanho do projeto?</p> <p>Jogador:</p> <p>EV.3.3.1. Você pode pedir para a equipe estimar o tempo em horas necessário para executar cada atividade do projeto.</p> <p>Feedback:</p> <p>EV.3.3.1.1. Tome cuidado! Quando falamos no tamanho não estamos pensando no tempo necessário para desenvolver um software ou uma tarefa. O tamanho é uma medida do volume de um software sob o ponto de vista do usuário. Pode ser determinado contando o número de tabelas do banco de dados, classes, objetos, relatórios, telas, consultas ao banco de dados, linhas de código fonte, etc. Pense mais um pouco e volte a falar com o gerente de projeto.</p> <p>EV.3.3.2. Primeiro você deve quebrar as tarefas do projeto em atividades menores mais fáceis de serem estimadas.</p> <p>Feedback:</p> <p>EV.3.3.2.1. Muito bem! Com a decomposição das tarefas do projeto em atividades menores é possível visualizar melhor o que deve ser feito em cada tarefa, permitindo que se tenham estimativas mais realísticas.</p>	
<p>Personagem:</p> <p>EV.3.4. E depois de quebrar as tarefas o que eu faço?</p> <p>Jogador:</p> <p>EV.3.4.1. Ai você deve determinar a complexidade de cada atividade. Seria interessante você envolver a equipe nesta tarefa.</p> <p>Feedback:</p> <p>EV.3.4.1.1. Boa idéia. No nível G, a estimativa de tamanho pode ser feita baseada na complexidade das atividades, você não precisa utilizar uma técnica muito complexa para isso, como a Análise de Pontos por Função.</p> <p>EV.3.4.2. Depois de quebrar as tarefas você já tem o tamanho do projeto, que será a quantidade de atividades do mesmo.</p> <p>Feedback:</p> <p>EV.3.4.2.1. Cuidado! Pois se você considerar que o tamanho do projeto será somente a quantidade de atividades do mesmo, você pode ter projetos com o mesmo número de atividades, mas com durações completamente diferentes, isso pode ocorrer por que as atividades de um projeto podem ser muito mais complexas que as de outro. Pense mais um pouco e depois converse com o gerente de projetos novamente.</p>	GPR3 = T
<p>Personagem:</p> <p>EV.3.5. Entendi! E depois de determinar o tamanho do projeto eu já posso elaborar o cronograma do projeto, certo?</p> <p>Jogador:</p> <p>EV.3.5.1. Ainda não! Primeiro você deve determinar o esforço de cada atividade.</p> <p>Feedback:</p>	GPR4 = P

<p>EV.3.5.1.1. Isso mesmo! Para poder elaborar o cronograma você precisa antes saber a duração das atividades.</p> <p>EV.3.5.2. Isso mesmo! Se você já determinou o tamanho das atividades, você já pode elaborar o cronograma.</p> <p>Feedback:</p> <p>EV.3.5.2.1. Pense bem, como você vai elaborar o cronograma se você ainda não determinou a duração das atividades? O tamanho das atividades não determina a duração das mesmas e sim o volume de trabalho dessas atividades. Para poder elaborar o cronograma você deve antes determinar o esforço de cada atividade que é justamente a duração das mesmas. Revise sua resposta e volte a falar com o gerente de projetos novamente.</p>	
<p>Personagem:</p> <p>EV.3.6. Claro, como eu pude esquecer o esforço. Você poderia me dar um exemplo de como eu posso estimar o esforço das atividades?</p> <p>Jogador:</p> <p>EV.3.6.1. Você pode derivar o esforço das atividades a partir do tamanho das mesmas utilizando os dados históricos.</p> <p>Feedback:</p> <p>EV.3.6.1.1. Essa pode ser uma boa opção para o futuro, pois hoje a empresa ainda não possui dados históricos. Pense um pouco mais e depois volte a conversar com o gerente de projetos.</p> <p>EV.3.6.2. Você pode pedir para equipe estimar o esforço de cada atividade com base na complexidade identificada anteriormente.</p> <p>Feedback:</p> <p>EV.3.6.2.1. Ótima idéia. Envolvendo a equipe nas estimativas você deve ter estimativas mais precisas, pois elas serão realizadas pelas pessoas que executarão as atividades. No futuro, dados históricos poderão apoiar a estimativa de esforço.</p>	GPR4 = L
<p>Personagem:</p> <p>EV.3.7. Certo! Eu não entendi muito bem o que são dados históricos, você poderia me explicar melhor?</p> <p>Jogador:</p> <p>EV.3.7.1. Dados históricos são informações de custo, esforço e tempo de projetos executados anteriormente, além de dados apropriados de escala para equilibrar as diferenças de tamanho e complexidade dos mesmos.</p> <p>Feedback:</p> <p>EV.3.7.1.1. Isso mesmo! Lembre-se que você deve guardar essas informações para que em projetos futuros possamos derivar as <u>estimativas</u> de esforço e custo através dos dados histórico aplicados sobre o tamanho dos projetos.</p> <p>EV.3.7.2. Dados históricos são todos os artefatos produzidos em projetos passados, como por exemplo, plano de projeto, documento de requisitos, atas de reunião, etc.</p> <p>Feedback:</p>	GPR4 = L

<p>EV.3.7.2.1. Você até pode considerar que os artefatos produzidos em projetos passados como dados históricos, mas isso pode dificultar a localização de informações depois. Por exemplo, se você deseja obter a estimativa de esforço de um projeto a partir da estima de esforço de projetos passados de mesmo tamanho, você teria que abrir o plano de projeto de todos os projetos passados para verificar quais projetos possuem o mesmo tamanho que o projeto em questão e verificar qual foi a estimativa de esforço destes projetos. O ideal é que você verifique quais informações podem ser utilizadas em projetos futuros e monte um banco de dados de informações de projetos passados para facilitar a obtenção destas informações depois. Pense um pouco sobre isso e volte a conversar com o gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.3.8. Entendi! Você mencionou custos e eu fiquei com uma dúvida, eu preciso determinar o custo das atividades também?</p> <p>Jogador:</p> <p>EV.3.8.1. Depende. Se você tiver que gerenciar os custos financeiros do projeto, então você precisará pensar no valor em unidades monetárias.</p> <p>Feedback:</p> <p>EV.3.8.1.1. Muito bem! É isso mesmo, o gerente de projetos não é obrigado a determinar os custos do projeto em unidades monetárias. Ele só irá fazer isso se ele precisar gerenciar os custos financeiros do projeto.</p> <p>EV.3.8.2. Sim. Sempre teremos que gerenciar os custos do projeto em valores monetários.</p> <p>Feedback:</p> <p>EV.3.8.2.1. Errado! Nos casos em que o gerente de projetos não irá gerenciar os custos financeiros do projeto, ele não precisa determinar o custo das atividades em unidades monetárias. Revise sua resposta e volte a conversar com o gerente de projetos.</p>	GPR4 = L
<p>Personagem:</p> <p>EV.3.9. Ok, mas como eu não trabalharei o aspecto financeiro do projeto, então eu não gerenciarei custos, certo?</p> <p>Jogador:</p> <p>EV.3.9.1. Irá sim, mas indiretamente. No nosso caso, podemos considerar que o custo de cada atividade será o esforço planejado para ela. Ou seja, nossa moeda será a unidade de tempo.</p> <p>Feedback:</p> <p>EV.3.9.1.1. Isso mesmo! Mas lembre-se que apesar do gerente de projetos não trabalhar com valores monetários, é preciso deixar claro isso no plano do projeto ou na política organizacional, e também que os custos serão tratados como esforço.</p> <p>EV.3.9.2. Você tem razão. Não precisaremos nos preocupar com valores financeiros, ou seja, custos das tarefas.</p> <p>Feedback:</p>	GPR4 = T

<p>EV.3.9.2.1. Tome cuidado! Mesmo que o gerente de projetos não trabalhe com valores financeiros, o custo precisa ser determinado. Neste caso o custo de cada atividade será o esforço planejado para a mesma. Reflita sobre isso e volte a conversar com o gerente de projetos.</p>	
<p>Personagem: EV.3.10. Ok chefe. E depois de definir o esforço das tarefas qual será o próximo passo?</p>	GPR5 = P GPR17 = P
<p>Jogador: EV.3.10.1. O próximo passo será elaborar o cronograma do projeto, identificando também os marcos do mesmo.</p> <p>Feedback: EV.3.10.1.1.Isto! Para elaborar o cronograma do projeto é importante que você saiba quais são os marcos e as datas desses marcos. Além disso, não esqueça que você precisará estimar uma data para o encerramento do projeto.</p>	
<p>EV.3.10.2. O próximo passo será elaborar o cronograma somente da primeira iteração.</p> <p>Feedback: EV.3.10.2.1.Cuidado! Durante o planejamento do projeto é importante que você elabore o cronograma para todo o projeto. Embora o detalhamento possa ser feito a cada iteração, você deve ter uma estimativa de quando o projeto terminará. Revise sua escolha e volte a conversar com o gerente de projetos.</p>	
<p>Personagem: EV.3.11. Marco? Eu já ouvi este termo, mas tenho dificuldades em saber o que é um marco em um projeto, você poderia me dar alguns exemplos de marco.</p>	GPR5 = P GPR17 = L
<p>Jogador: EV.3.11.1. Término da fase de planejamento, término de uma iteração, entrega de uma versão prévia do produto para o cliente, data em que um determinado recurso estará disponível, ou seja, qualquer evento significativo no projeto deve ser considerado um marco.</p> <p>Feedback: EV.3.11.1.1.Isto mesmo! Marcos é qualquer evento importante no projeto e não necessariamente o cliente precisa estar envolvido.</p>	
<p>EV.3.11.2. É qualquer evento no projeto em que o cliente será envolvido, por exemplo: validação dos requisitos do projeto, apresentação do plano de projeto, entrega de uma versão parcial do produto, entrega da versão final do produto, etc.</p> <p>Feedback: EV.3.11.2.1.Cuidado! Os exemplos que você deu realmente são marcos, mas um marco não necessariamente precisa envolver o cliente. Pense um pouco mais e volte a conversar com o gerente de projetos.</p>	
<p>Personagem: EV.3.12. Entendi, mas por que a determinação de marcos no projeto é importante?</p>	GPR5 = P GPR11 = P GPR17 = T

<p>Jogador:</p> <p>EV.3.12.1. Para que a equipe possa saber os eventos importantes que ocorrerão durante o projeto e o que eles devem entregar nesses eventos.</p> <p>Feedback:</p> <p>EV.3.12.1.1. Isso também é importante, mas o mais importante nos marcos é verificar o andamento do projeto de forma mais ampla como por exemplo, se os entregáveis que deveriam ser entregues foram entregues, a qualidade desses entregáveis ou motivo da não entrega, etc. E caso seja necessário estas revisões devem questionar inclusive a viabilidade de continuidade do projeto. Reflita um pouco mais sobre sua resposta e volte a conversar com o gerente de projetos.</p> <p>EV.3.12.2. É importante para que possamos verificar de forma mais ampla o andamento do projeto e a viabilidade de continuarmos o projeto.</p> <p>Feedback:</p> <p>EV.3.12.2.1. Correto! Revisões devem ser realizadas nos marcos visando verificar de forma mais ampla o andamento do projeto, como por exemplo, se os entregáveis que deveriam ser entregues foram entregues, a qualidade desses entregáveis ou motivo da não entrega, etc. E caso seja necessário estas revisões devem questionar inclusive a viabilidade de continuidade do projeto.</p>	
<p>Personagem:</p> <p>EV.3.13. Certo chefe! Então acho que agora eu já posso planejar nosso primeiro projeto, pois eu já sei tudo o que eu preciso fazer.</p> <p>Jogador:</p> <p>EV.3.13.1. Calma, ainda não definimos como faremos o orçamento dos projetos.</p> <p>Feedback:</p> <p>EV.3.13.1.1. Bem lembrado! O orçamento é importante para que você tenha o controle dos gastos no projeto.</p> <p>EV.3.13.2. Acho que pode sim, já cobrimos todas as etapas necessárias para o planejamento do projeto.</p> <p>Feedback:</p> <p>EV.3.13.2.1. Pense bem, mesmo que o gerente de projeto não trabalhe com valores financeiros, você não acha que é preciso listar tudo que será necessário ser adquirido, além das horas da equipe para que o financeiro possa ter um controle das finanças do projeto? Pense nisso e volte a conversar com o gerente de projeto.</p>	GPR5 = L
<p>Personagem:</p> <p>EV.3.14. Orçamento? Mas eu não irei trabalhar com valores financeiros, isso não é responsabilidade do setor financeiro?</p> <p>Jogador:</p> <p>EV.3.14.1. Você tem razão, esqueça o orçamento, deixe isso para o setor financeiro.</p> <p>Feedback:</p> <p>EV.3.14.1.1. Pense bem, mesmo que o gerente de projeto não trabalhe com</p>	GPR5 = L

<p>valores financeiros, você não acha que é preciso listar tudo que será necessário ser adquirido, além das horas da equipe para que o financeiro possa ter um controle das finanças do projeto? Pense nisso e volte a conversar com o gerente de projeto.</p> <p>EV.3.14.2. Sim, mas apesar de você não trabalhar com valores financeiros, você precisa listar tudo que será necessário ser adquirido, além das horas da equipe para que o financeiro possa ter um controle das finanças do projeto.</p> <p>Feedback:</p> <p>EV.3.14.2.1. É isso mesmo, mesmo que o gerente de projetos não trabalhe com valores financeiros, é preciso listar todos os recursos físicos que possa ser necessário ser adquirido, além de todas as horas do projeto para que o financeiro tenha o controle das finanças do projeto.</p>	
<p>Personagem:</p> <p>EV.3.15. Entendi! Acho que o planejamento de nossos projetos ficará bem completo dessa forma, você acha que ainda está faltando alguma coisa?</p> <p>Jogador:</p> <p>EV.3.15.1. Acho que não está faltando mais nada, podemos começar a planejar nosso primeiro projeto.</p> <p>Feedback:</p> <p>EV.3.15.1.1. Como não falta mais nada? Você já pensou em como você vai planejar os riscos do projeto? E os dados do projeto? E os recursos físicos? Pense um pouco mais sobre sua resposta e volte a conversar com o gerente de projetos.</p> <p>EV.3.15.2. Acho que ainda faltam algumas coisas, mas vamos começar nosso primeiro projeto até para podermos validar o que fizemos até agora, e depois melhoraremos nosso planejamento.</p> <p>Feedback:</p> <p>EV.3.15.2.1. Excelente resposta. Apesar de ainda estar faltando considerar algumas coisas no planejamento, como riscos, dados, recursos humanos e infra-estrutura, é bom que você já comece o seu primeiro projeto para avaliarmos se estamos no caminho certo.</p>	GPR6 = P GPR8 = P GPR9 = P
<p>Personagem:</p> <p>EV.3.16. Ok chefe, então eu vou começar a planejar nosso primeiro projeto.</p> <p>Jogador:</p> <p>EV.3.16.1. Ok! Pode começar. Só não se esqueça de revisar o plano de projeto com a equipe quando você terminar o planejamento e de monitorar o plano durante a execução do projeto.</p> <p>Feedback:</p> <p>EV.3.16.1.1. Muito bem! A revisão do plano de projeto com a equipe é importante para obter o comprometimento deles com o que está no plano. Nessa revisão eles devem verificar se eles concordam com o que está no plano, caso eles não concordem o plano deve ser ajustado até que todo mundo concorde com o mesmo. Depois que o projeto estiver em andamento o plano de projeto deve ser monitorado para verificar se ele está sendo executado</p>	GPR12 = L GPR13 = P GPR14 = P GPR15 = p

<p>conforme planejado e corrigir eventuais inconsistências.</p> <p>EV.3.16.2. Isso! Pode começar. Só não se esqueça de me comunicar quando você terminar o planejamento.</p> <p>Feedback:</p> <p>EV.3.16.2.1. A comunicação no projeto é importante, mas você deve lembrar que é necessário revisar o plano de projeto com a equipe depois que o planejamento for finalizado para o obter o comprometimento da mesma com o plano. Revise sua resposta e volte a conversar com o gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.3.17. Deixa comigo chefe. Muito obrigado pela ajuda!</p> <p>Jogador:</p> <p>EV.3.17.1. De nada! Quando você precisar é só me chamar.</p>	

Tabela 33. Evento 4 - pedido de alteração de requisitos

EV.4. Pedido de alteração de requisitos.	
Descrição	O gerente de produto ligou pedindo uma alteração num dos requisitos do projeto. Eu repassei este pedido para a equipe. Converse com eles sobre a situação deste pedido.
Condição	Um dia após o evento 3.
Personagem	Desenvolvedores.
Ação	Conversar.
Diálogos	Resultados Esperados
<p>Personagem:</p> <p>EV.4.1. Oi Chefe, você viu o pedido de alteração (de requisito) do nosso gerente de produto?</p> <p>Jogador:</p> <p>EV.4.1.1. Ainda não.</p>	
<p>Personagem:</p> <p>EV.4.2. Ele nos pediu para retirar um dos campos da tela de cadastro. É muito tranquilo. Devo fazer bem rápido. Posso fazer agora e já me livrar?</p> <p>Jogador:</p> <p>EV.4.2.1. Ótimo. Parece mesmo bem tranquilo. Faça a alteração e me avise quando ela estiver pronta. Bom trabalho!</p> <p>Feedback:</p> <p>EV.4.2.1.1. Tome cuidado! Mesmo que a alteração pareça tranquila você precisa formalizar esse pedido de alteração para que possamos ter histórico das mudanças no projeto. Revise sua resposta e volte a conversar com o gerente de produto.</p> <p>EV.4.2.2. A alteração está registrada, usando nosso template (modelo de documento)</p>	GRE5 = P

<p>de pedido de alteração?</p> <p>Feedback:</p> <p>EV.4.2.2.1. Isso mesmo! Mesmo que a alteração pareça tranquila você precisa formalizar esse pedido de alteração para que possamos ter histórico das mudanças no projeto.</p>	
<p>Personagem:</p> <p>EV.4.3. Achei que a alteração seria tão simples que não valeria a pena gerar um documento para registrá-la.</p> <p>Jogador:</p> <p>EV.4.3.1. Certo, mas se não registrarmos o pedido de mudança pode ser que tenhamos problemas depois, pois o documento de requisitos aprovado não irá bater com as funcionalidades do sistema.</p> <p>Feedback:</p> <p>EV.4.3.1.1. Ótimo argumento! O registro das solicitações de mudanças permite que se tenha um histórico das mudanças nos requisitos, evitando assim problemas durante a validação do sistema.</p> <p>EV.4.3.2. Mas lembre-se que nós estamos implementando o modelo MPS.BR e o modelo exige que todos os pedidos de mudanças sejam registrados.</p>	<p>GRE1 = P GRES5 = P</p>
<p>Personagem:</p> <p>EV.4.4. Certo, vou registrar a alteração de acordo com nosso processo interno. Depois disso, posso executar a mudança?</p> <p>Jogador:</p> <p>EV.4.4.1. Sim. Já teremos a garantia da documentação do pedido de alteração. Além disso, você já avaliou que o impacto é pequeno.</p> <p>Feedback:</p> <p>EV.4.4.1.1. Atenção! Mesmo que o impacto de uma solicitação de mudança seja pequeno, é importante formalizar essa análise de impacto, identificando tudo que pode ser afetado pela mudança e registrando no template de solicitação de mudança. Revise sua resposta e volte a conversar com sua equipe.</p>	<p>GRES5 = L</p>
<p>EV.4.4.2. Ainda não. Você precisa formalizar a sua análise do impacto desta mudança. Isso não afetará alguma tabela do banco de dados? Quais os relatórios que serão afetados, uma vez que a informação não será mais cadastrada?</p> <p>Feedback:</p> <p>EV.4.4.2.1. Isso mesmo! Mesmo que se saiba que o impacto de uma solicitação de mudança seja pequeno, é importante formalizar essa análise de impacto, para isso deve-se identificar tudo que pode ser afetado pela mudança e registrar no template de solicitação de mudança.</p>	

<p>Personagem:</p> <p>EV.4.5. Mas como saber tudo que será afetado?</p> <p>Jogador:</p> <p>EV.4.5.1. É só varrer o código do sistema e verificar tudo que é afetado. Tente fazer uma busca pelo nome do campo. Pode ajudar.</p> <p>Feedback:</p> <p>EV.4.5.1.1. A busca pelo nome do campo pode ajudar, mas você não conseguirá identificar tudo que será afetado somente revisando o código. Você deve possuir um mecanismo que permita identificar todos os relacionamentos com um requisito, incluindo tabelas, código fonte, roteiro de testes, etc. Pense mais um pouco e volte a conversar com sua equipe.</p> <p>EV.4.5.2. Calma, é para isso que implementamos nosso mecanismo de rastreabilidade.</p> <p>Feedback:</p> <p>EV.4.5.2.1. Perfeito! A rastreabilidade é um excelente mecanismo para identificar tudo que será afetado por uma mudança, pois através dela, você conseguirá visualizar todos os relacionamentos com o requisito alterado, incluindo tabelas, código fonte, roteiros de teste, etc.</p>	<p>GRE3 = L GRE5 = L</p>
<p>Personagem:</p> <p>EV.4.6. Ufa! Tinha esquecido dessa rastreabilidade e olha que você falou bastante dela no nosso treinamento sobre o processo. E depois desta análise?</p> <p>Jogador:</p> <p>EV.4.6.1. Precisamos avaliar o impacto no custo e no esforço do projeto. Com base nisso, iremos verificar se haverá necessidade de replanejarmos o projeto com novas datas e com a revisão dos custos. Se não houver, iremos pedir apenas a aprovação da alteração.</p> <p>Feedback:</p> <p>EV.4.6.1.1. Isso mesmo! É muito importante que você solicite a aprovação formal da mudança para o gerente de produto, para que depois ele não diga que não pediu nada do que foi feito.</p> <p>EV.4.6.2. A partir desta análise, devemos avaliar o impacto no custo e no esforço do projeto, e replanejar o mesmo com novas datas e com a revisão dos custos. Depois disso você já pode executar a mudança.</p> <p>Feedback:</p> <p>EV.4.6.2.1. Tome cuidado, antes de executar a mudança você deve pedir a aprovação formal do gerente de produto, para que depois ele não diga que não pediu nada do que foi feito. Pense mais um pouco e volte a conversar com sua equipe.</p>	<p>GRE5 = L GPR5 = T</p>
<p>Personagem:</p> <p>EV.4.7. E se ele não aprovar?</p> <p>Jogador:</p> <p>EV.4.7.1. Iremos tomar mais cuidado no futuro para evitar pedidos de alteração dos requisitos já aprovados.</p>	<p>GRE1 = P GRE5 = L</p>

<p>Feedback:</p> <p>EV.4.7.1.1. Errado! Pedidos de mudança sempre irão ocorrer, o que você pode fazer é planejar um tempo no seu cronograma para análise de solicitações de mudança, assim o seu projeto não será afetado pela análise de solicitações de mudanças. Pense nisso e volte a conversar com sua equipe.</p> <p>EV.4.7.2. Não tem problema. Eu já planejei um buffer no nosso cronograma com um bloco de horas para ser gasto com análise de pedidos de mudança.</p> <p>Feedback:</p> <p>EV.4.7.2.1. Muito bem! Esse buffer fará com que o projeto em desenvolvimento não seja afetado pela análise de solicitações de mudanças.</p>	
<p>Personagem:</p> <p>EV.4.8. E se ele aprovar? Podemos iniciar a alteração?</p> <p>Jogador:</p> <p>EV.4.8.1. Sim. Esta documentação será suficiente para mantermos um histórico das alterações.</p> <p>Feedback:</p> <p>EV.4.8.1.1. Não, se a alteração for aprovada você precisa ainda revisar todos os artefatos do projeto para que eles estejam consistentes com a mudança aprovada. Revise sua resposta e volte a conversar com sua equipe.</p> <p>EV.4.8.2. Antes, precisamos revisar o plano para garantir que todos os ajustes nos artefatos impactados sejam tratados.</p> <p>Feedback:</p> <p>EV.4.8.2.1. Isso! É muito importante que os artefatos afetados pela mudança sejam ajustados para evitar problemas na hora da validação do sistema. Não se esqueça também de atualizar a rastreabilidade, se ela for afetada pela mudança.</p>	<p>GRE3 = T GRE4 = L GRE5 = L</p>
<p>Personagem:</p> <p>EV.4.9. Mas, se alterarmos o plano, não precisaremos de um novo kickoff do projeto?</p> <p>Jogador:</p> <p>EV.4.9.1. Sim. Assim que o plano estiver revisado, você poderá reunir toda a equipe e repassar todos os requisitos novamente. Assim, teremos um novo comprometimento de todos com o plano e com os requisitos.</p> <p>Feedback:</p> <p>EV.4.9.1.1. Apesar de ser importante que você revise novamente o plano com a equipe após a alteração, você não precisa revisar todo o plano, e sim só a parte afetada e com as pessoas que serão afetadas. Pense mais um pouco e volte a conversar com sua equipe.</p> <p>EV.4.9.2. Não precisa ser um kickoff. Você pode apresentar as mudanças apenas para aqueles que serão afetados. Não esqueça de verificar se as estimativas estão adequadas.</p> <p>Feedback:</p> <p>EV.4.9.2.1. Isso mesmo! As pessoas que serão afetadas pelas mudanças</p>	<p>GPR12 = T GRE2 = P GRE5 = T</p>

<p>precisam se comprometer novamente com o plano, para isso você deve apresentar o plano para as mesmas e verificar se elas concordam com as estimativas, e caso elas não concordem o plano precisa ser ajustado. Lembre-se que esta apresentação do plano para as pessoas afetadas deve ser registrada de alguma forma, pode ser através de uma ata de reunião, para que se tenham evidências de que as pessoas afetadas pela mudança se comprometeram com a mesma.</p>	
<p>Personagem:</p> <p>EV.4.10. Ok Chefe. Acho que nosso processo tá ficando bem legal. Realmente, com as alterações registradas e com a análise de impacto baseada na rastreabilidade, não teremos surpresas na entrega final. Obrigado pela ajuda!</p>	
<p>Jogador:</p> <p>EV.4.10.1. De nada.</p>	

Tabela 34. Evento 5 - definição do planejamento de riscos

EV.5. Definição do planejamento de riscos.	
Descrição	O gerente de projetos quer conversar com você para definir como o planejamento de riscos dos projetos será realizado.
Condição	Um dia após o evento 4.
Personagem	Gerente de projetos.
Ação	Conversar.
Diálogos	Resultados Esperados
Personagem:	GPR6 = P
EV.5.1. Oi Chefe! Em nossa última reunião você falou que precisamos pensar em como vamos planejar os riscos em nossos projetos, você já sabe como vamos fazer isso?	
Jogador:	
EV.5.1.1.Sim, temos que implementar um processo de gerenciamento de riscos completo, com ações mitigatórias e de contigência para cada risco, para que possamos controlar melhor os riscos em nossa empresa e além disso isso é uma exigência do nível G do MPS.BR.	
Feedback:	
EV.5.1.1.1. Atenção, para o nível G você não precisa ter um Gerenciamento de Riscos, ou seja, a identificação, o gerenciamento e a redução contínua dos riscos nos níveis organizacionais e de projeto. No nível G, os riscos são acompanhados para verificar como afetam o projeto e para se tomar ações, mesmo que ainda sem um gerenciamento completo. Pense um pouco mais e volte a conversar com o gerente de projetos.	
EV.5.1.2.Sei sim, para facilitar a identificação dos riscos, vamos elaborar uma planilha com problemas que já tivemos no passado e você irá examinar quais destes problemas poderão acontecer novamente no projeto.	
Feedback:	

<p>EV.5.1.2.1. Muito bem! Essa planilha irá facilitar e acelerar a identificação dos riscos do projeto, mas mantenha esta planilha atualizada. Porém, pense também em novos problemas que poderão ocorrer no projeto (não olhe somente para o passado).</p> <p>EV.5.1.3.Ainda não.</p> <p>Feedback:</p> <p>EV.5.1.3.1. Ok, pense um pouco mais e volte a conversar com o gerente de projetos mais tarde.</p>	
<p>Personagem:</p> <p>EV.5.2. E depois de identificar os riscos, o que eu faço?</p> <p>Jogador:</p> <p>EV.5.2.1.Precisamos registrar cada risco na nossa planilha de riscos e pensar na sua probabilidade de ocorrência e impacto no projeto caso ele ocorra.</p> <p>Feedback:</p> <p>EV.5.2.1.1. Muito bem! É importante lembrar também que a probabilidade de ocorrência e o impacto de um risco já conhecido deve ser avaliada para cada projeto, pois estas informações podem variar de projeto para projeto.</p> <p>EV.5.2.2.Neste momento, iremos apenas registrar os riscos na nossa planilha de riscos.</p> <p>Feedback:</p> <p>EV.5.2.2.1. Cuidado! Somente registrar os riscos não é suficiente, você deve pensar também na probabilidade de ocorrência e no impacto de cada risco e colocar nessa planilha. Revise sua resposta e volte a conversar com o gerente de projetos.</p>	GPR6 = L
<p>Personagem:</p> <p>EV.5.3. Mas, por que precisamos avaliar a probabilidade e o impacto de cada risco?</p> <p>Jogador:</p> <p>EV.5.3.1.Pois, a partir destas informações, podemos verificar o grau de exposição que nosso projeto está em relação ao risco. A partir deste fator de exposição, podemos priorizar os riscos.</p> <p>Feedback:</p> <p>EV.5.3.1.1. Isso mesmo! É importante priorizar os riscos por que após a identificação dos riscos é possível que se tenha uma lista numerosa, e normalmente não é possível realizar ações para tratar e monitorar todos os riscos, sendo importante escolher um subconjunto. Além disso, essas informações podem ser utilizadas para apoiar a análise de viabilidade do projeto.</p> <p>EV.5.3.2.Por que estas informações serão importantes para os próximos projetos, oferecendo dados históricos.</p> <p>Feedback:</p> <p>EV.5.3.2.1. Pense bem! Embora seja adequado guardar as informações sobre riscos, conhecer a chance de um risco ocorrer e seu impacto não deve ser</p>	GPR6 = L

<p>somente para dados históricos. Você deve utilizar estas informações para avaliar a prioridade do risco e apoiar a análise da viabilidade do projeto. Reflita sobre isso e volte a conversar com o gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.5.4. Entendi! Precisamos pensar em ações para tratar os riscos também?</p> <p>Jogador:</p> <p>EV.5.4.1. Não, pois o nível G não exige a definição de respostas aos riscos.</p> <p>Feedback:</p> <p>EV.5.4.1.1. Cuidado! Você não deve pensar em melhoria só para atender a um modelo, mas sim para melhorar sua forma de trabalho. Realmente, o nível G não exige que você planeje ações para tratar os riscos, mas pense se talvez não seja adequado você pensar em ações pelo menos para os riscos mais críticos, dessa forma você saberá o que fazer quando esse risco ocorrer. Pense nisso, e volte a conversar com o gerente de projetos.</p> <p>EV.5.4.2. Podemos pensar em ações para aqueles riscos com maior prioridade.</p> <p>Feedback:</p> <p>EV.5.4.2.1. Isso mesmo! Você pode pensar em ações para os riscos mais críticos, desta forma você saberá o que fazer quando o risco ocorrer. Mas vale lembrar que neste momento não é obrigatório que você pense nas ações de respostas aos riscos durante o planejamento, você pode pensar nas ações quando os riscos ocorrerem.</p>	GPR6 = L
<p>Personagem:</p> <p>EV.5.5. Ok chefe. Depois de registrar todos os riscos, eu tenho que fazer mais alguma coisa?</p> <p>Jogador:</p> <p>EV.5.5.1. Neste momento, o foco principal é podermos demonstrar que os riscos foram registrados formalmente. O acompanhamento será realizado futuramente, quando tivermos mais maturidade.</p> <p>Feedback:</p> <p>EV.5.5.1.1. Como assim não precisa acompanhar? Não faz sentido identificarmos os riscos se não vamos acompanhá-los. Você deve acompanhar os riscos para verificar se a probabilidade ou impacto tiveram alguma alteração, verificar se existem novos riscos, eliminar aqueles riscos que não fazem mais sentido, e verificar se algum risco ocorreu. Pense mais um pouco e volte a conversar com o gerente de projetos.</p> <p>EV.5.5.2. Claro que sim. Durante a execução do projeto, devemos acompanhar os riscos para verificar se a probabilidade ou impacto tiveram alguma alteração, perceber novos riscos ou eliminar aqueles que não fazem mais sentido.</p> <p>Feedback:</p> <p>EV.5.5.2.1. É isso aí! Também é importante lembrar que o acompanhamento deve ser registrado de alguma forma, pode ser no próprio plano de projetos, ou em outro documento, como por exemplo, ata de reunião de acompanhamento.</p>	GPR6 = T GPR15 = P
Personagem:	GPR15 = L

<p>EV.5.6. E se um risco ocorrer o que eu devo fazer?</p> <p>Jogador:</p> <p>EV.5.6.1. Você pode registrar os problemas causados pelo risco e planejar ações corretivas para os problemas identificados, ou executar o plano de resposta ao risco caso o mesmo exista.</p> <p>Feedback:</p> <p>EV.5.6.1.1. Isso mesmo! No nível G não é obrigatório que você tenha um plano de resposta aos riscos, ao invés disso, você pode registrar os problemas causados pelos riscos e planejar ações junto com as partes interessadas, para corrigir os desvios em relação ao planejado e para prevenir a repetição dos problemas identificados.</p> <p>EV.5.6.2. Você deve ter um plano de resposta para os riscos que deve ser executado quando um risco ocorrer.</p> <p>Feedback:</p> <p>EV.5.6.2.1. Atenção! Ter um plano de resposta aos riscos é importante, mas neste momento ele não é obrigatório. Ao invés de ter um plano de resposta aos riscos, você pode registrar os problemas causados pelos riscos e planejar ações junto com as partes interessadas, para corrigir os desvios em relação ao planejado e para prevenir a repetição dos problemas identificados. Revise sua resposta e volte a conversar com o gerente de projetos.</p>	GPR18 = L GPR19 = L
<p>Personagem:</p> <p>EV.5.7. Além de planejar as ações corretivas eu devo fazer mais alguma coisa?</p> <p>Jogador:</p> <p>EV.5.7.1. Não, neste momento apenas o planejamento das ações corretivas é suficiente.</p> <p>Feedback:</p> <p>EV.5.7.1.1. Não, além de planejar as ações corretivas é importante que você monitore com certa frequência estas ações para verificar se elas já foram resolvidas, e caso não se consiga resolve-las neste nível de gerência, elas devem ser escalonadas a níveis superiores de gerência. Pense mais um pouco e volte a conversar com o gerente de projetos.</p> <p>EV.5.7.2. Sim, você deve acompanhar as ações corretivas até que elas sejam concluídas.</p> <p>Feedback:</p> <p>EV.5.7.2.1. É isso aí, as ações corretivas precisam ser acompanhadas com uma certa freqüência para verificar se elas já foram resolvidas, e caso não se consiga resolve-las neste nível de gerência, elas devem ser escalonadas a níveis superiores de gerência.</p>	GPR15 = T GPR18 = P GPR19 = P
<p>Personagem:</p> <p>EV.5.8. Ok chefe, então agora eu vou colocar isso que acabamos de definir em prática. Muito obrigado pela ajuda.</p> <p>Jogador:</p>	

EV.5.8.1.Ok. Bom trabalho!	
----------------------------	--

Tabela 35. Evento 6 - problema com os artefatos do projeto

EV.6. Problema com os artefatos do projeto.	
Descrição	O gerente de projetos percebeu que muitas vezes os membros da equipe têm dificuldade em achar os artefatos do projeto, e ainda às vezes utilizam os artefatos errados, causando uma série de transtornos durante os projetos. converse com o gerente de projetos e o ajude a encontrar uma solução para este problema.
Condição	Um dia após o evento 5.
Personagem	Gerente de projetos.
Ação	Conversar.
Diálogos	Resultados Esperados
Personagem: EV.6.1. Oi Chefe! Estava querendo mesmo falar com você, preciso da sua ajuda para resolver um problema.	
Jogador: EV.6.1.1.Que problema?	
Personagem: EV.6.2. Eu percebi que muitas vezes os membros da equipe têm dificuldade em achar os artefatos do projeto, e o pior é que às vezes utilizam os artefatos errados, causando uma série de transtornos durante os projetos.	GPR 9 = P
Jogador: EV.6.2.1.Isso é um típico problema de falta de planejamento de dados. Feedback: EV.6.2.1.1. Isso mesmo, a falta de planejamento de dados do projeto pode causar vários problemas como este, por exemplo: não encontrar um determinado artefato do projeto, não saber como gerar um artefato, utilizar um artefato errado, esquecer de produzir um artefato, etc. EV.6.2.2.Isso é um problema de comunicação, você precisa falar para os membros da equipe perguntarem mais, quando eles estiverem com alguma dúvida. Feedback: EV.6.2.2.1. Tome cuidado, veja que a comunicação não resolve completamente este tipo de problema, por exemplo: o que acontecerá se uma pessoa que possua o conhecimento sobre um artefato estiver ausente da empresa? Pense melhor na sua resposta e volte a conversar com o gerente de projetos.	
Personagem: EV.6.3. Dados? Você já falou sobre isso antes, mas eu não lembro muito bem o que são?	GPR9 = P

<p>Jogador:</p> <p>EV.6.3.1. Dado é qualquer informação sobre o projeto, como por exemplo, prazo, custo, cronograma, riscos, requisitos, etc.</p> <p>Feedback:</p> <p>EV.6.3.1.1. Não, quando eu falo em dados do projeto eu estou falando em todos os artefatos que serão gerados ou utilizados no projeto, como por exemplo, documento de requisitos, plano de projeto, relatórios, manuais, atas de reuniões, templates, lições aprendidas, etc. Revise sua resposta e volte a conversar com o gerente de projetos.</p> <p>EV.6.3.2. Dados são todas as documentações necessárias para apoiar um projeto, por exemplo: documento de requisitos, plano de projeto, relatórios, manuais, atas de reuniões, templates, lições aprendidas, etc.</p> <p>Feedback:</p> <p>EV.6.3.2.1. Isso mesmo! Lembre-se que os dados podem estar em qualquer formato e existir em qualquer meio, como: impressos ou desenhados em diversos materiais; fotografias; meio eletrônico; e multimídia. Alguns dados podem ser disponibilizados aos clientes, enquanto outros não necessariamente o serão. A distribuição pode ocorrer de várias formas, incluindo a transmissão eletrônica.</p>	
<p>Personagem:</p> <p>EV.6.4. Entendi! Mas por que é importante planejar os dados do projeto?</p> <p>Jogador:</p> <p>EV.6.4.1. Para que os membros da equipe saibam onde encontrar os artefatos que serão utilizados no projeto.</p> <p>Feedback:</p> <p>EV.6.4.1.1. Veja bem, o planejamento de dados não é só para isso, você deve planejar os dados para poder controlar se todos os artefatos que deveriam ser gerados no projeto, foram gerados, armazenados e distribuídos de maneira correta. Pense mais um pouco e volte a conversar com o gerente de projetos.</p> <p>EV.6.4.2. É importante identificar os dados relevantes do projeto, para depois coletá-los, armazená-los e distribuí-los de forma controlada.</p> <p>Feedback:</p> <p>EV.6.4.2.1. Muito bem! Com o planejamento dos dados você saberá quais artefatos devem ser gerados durante o projeto. Mas lembre-se que durante a execução do projeto você deve monitorar se esses artefatos estão sendo coletados, armazenados e distribuídos de maneira correta. É importante lembrar também que a coleta dos dados implica em custos, assim eles devem ser coletados somente quando for necessário.</p>	GPR9 = P GPR14 = P
<p>Personagem:</p> <p>EV.6.5. Certo! E como eu posso planejar esses dados?</p> <p>Jogador:</p> <p>EV.6.5.1. Você pode elaborar uma planilha organizacional contendo artefatos que são comuns em todos os projetos, e durante o planejamento selecionar os artefatos que</p>	GPR9 = L

<p>serão gerados ou utilizados no projeto em questão.</p> <p>Feedback:</p> <p>EV.6.5.1.1. Perfeito. Isso irá acelerar a identificação dos artefatos que serão gerados ou utilizados no projeto, mas não se esqueça de verificar se para o projeto em questão não vai ser necessário algum outro artefato além desses que estão nessa planilha. Não se esqueça também de manter essa planilha organizacional atualizada, atualizando-a sempre que algum novo artefato comum a todos os projetos for identificado ou quando algum artefato não faz mais sentido.</p> <p>EV.6.5.2. Para o primeiro projeto você pode fazer uma reunião com a equipe para identificar todos os artefatos que serão gerados ou utilizados no projeto, depois é só copiar essas informações para os próximos projetos, pois irão ser sempre iguais.</p> <p>Feedback:</p> <p>EV.6.5.2.1. Cuidado! Concordo que a maioria dos artefatos serão os mesmos em todos os projetos, mas você não pode garantir que serão sempre os mesmos. Você até pode copiar essa informação de projetos passados, mas você deve verificar também se para o projeto em questão não será necessário algum artefato adicional. Pense mais um pouco e volte a conversar com o gerente de projetos.</p>	
<p>Personagem:</p> <p>EV.6.6. Entendi, mas que informações eu devo colocar nessa planilha organizacional?</p> <p>Jogador:</p> <p>EV.6.6.1. Identificador do dado, descrição do dado, local onde o dado é armazenado, descrição sobre como o dado é gerado, descrição sobre como o dado é acessado, responsável pela criação e manutenção do dado, quem tem acesso a este dado e qual o tipo de acesso (leitura/escrita), etc.</p> <p>Feedback:</p> <p>EV.6.6.1.1. Isso mesmo! Lembre-se também de especificar a periodicidade que o dado será gerado, para que você possa controlar se os dados estão sendo gerados na frequência correta, veja que essa informação pode ser diferente de projeto para projeto, então não basta copiar essa informação de projetos passados. Controlar a periodicidade que os dados serão gerados é importante, pois a geração desses dados implica em custos.</p> <p>EV.6.6.2. Neste momento basta colocarmos a descrição do dado e o local onde ele será armazenado, quando tivermos mais maturidade implementaremos um mecanismo de controle de acesso aos dados.</p> <p>Feedback:</p> <p>EV.6.6.2.1. Veja bem, concordo que você não precisa implementar um mecanismo de controle de acesso aos dados agora, mas você deve pensar pelo menos em quem pode ter acesso aos dados e qual o tipo de acesso, para que a equipe saiba quem tem acesso a cada artefato do projeto. Revise sua resposta e volte a conversar com o gerente de projetos.</p>	<p>GPR9 = L</p>
<p>Personagem:</p> <p>EV.6.7. Certo, tem mais alguma coisa que eu preciso colocar no planejamento dos</p>	<p>GPR9 = T GPR16 = P</p>

<p>dados?</p> <p>Jogador:</p> <p>EV.6.7.1.Sim, você deve pensar também na gerência de configuração desses dados, para que possamos ter um maior controle sobre as diferentes versões dos artefatos produzidos no projeto, além disso, isso é uma exigência do nível G do MPS.BR.</p> <p>Feedback:</p> <p>EV.6.7.1.1. Tome cuidado, lembre-se que no nível G você não é obrigado a implementar a Gerência de Configuração, mas você pode implementar se isso for uma necessidade de sua empresa. Revise sua resposta e converse novamente com o gerente de projetos.</p> <p>EV.6.7.2.Podemos integrar o plano de dados com o plano de comunicação adicionando as colunas periodicidade e meio de comunicação.</p> <p>Feedback:</p> <p>EV.6.7.2.1. Ótimo. A periodicidade é importante para saber de quanto em quanto tempo o dado será atualizado, e o meio de comunicação indica como as partes interessadas serão comunicadas quando for realizada uma atualização em um dado do projeto. Essa comunicação pode ocorrer de várias formas, como por exemplo: email, intranet, carta, etc. Lembre-se que durante o monitoramento dos dados do projeto deve-se verificar se os envolvidos estão sendo comunicados de maneira correta.</p>	
<p>Personagem:</p> <p>EV.6.8. Ok Chefe! Acho que o planejamento dos dados irá nos ajudar bastante mesmo. Agora eu vou colocar isso que nós discutimos em prática. Muito obrigado pela ajuda.</p> <p>Jogador:</p> <p>EV.6.8.1.Não tem por onde.</p>	

APÊNDICE C – TERMO DE CONCORDÂNCIA

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Você está sendo convidado para participar da pesquisa **Avaliação do Jogo Educacional SPI CITY**.

Você foi selecionado por possuir conhecimentos em Engenharia de Software e sua participação não é obrigatória.

O objetivo deste estudo é avaliar o efeito da aprendizagem no uso do jogo educacional SPI CITY.

Sua participação nesta pesquisa consistirá em utilizar o jogo educacional SPI CITY, responder questionários e avaliações.

Não foram identificados quaisquer riscos na participação deste experimento.

Você não terá qualquer despesa com a participação na pesquisa, e, igualmente não receberá nenhum valor em dinheiro para tanto.

Os benefícios relacionados com a sua participação são aprendizagem de conceitos, técnicas e práticas de melhoria de processos relacionadas ao modelo MPS.BR.

As informações obtidas através dessa pesquisa serão confidenciais e asseguramos o sigilo sobre sua participação. Os dados não serão divulgados de forma a possibilitar sua identificação.

Algumas fotografias poderão ser feitas como registro desta atividade, e poderão ser publicadas somente dentro do contexto desta pesquisa para comprovar a realização desta atividade, não podendo ser publicadas para nenhum outro propósito.

Você é responsável pelas informações prestadas e livre para interromper a participação na pesquisa a qualquer momento, sem a necessidade de justificar a decisão tomada e isso não afetará sua relação com o pesquisador.

José Luiz Silveira

Nome e assinatura do pesquisador

Declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa e concordo em participar.

Nome e assinatura do participante da pesquisa

APÊNDICE D – QUESTIONÁRIO DE PERFIL

Questionário de Perfil do Participante

Nome: _____.

Prezado participante, em relação ao seu perfil, responda as seguintes perguntas:

1) Você é acadêmico ou graduado no curso de:

[] B.Sc. em Ciência da Computação.

[] B.Sc. em Engenharia da Computação.

[] B.Sc. em Sistemas de Informação.

[] B.Sc. em outro curso na área de Computação. Qual?:_____.

[] B.Sc. em outro curso. Qual?:_____.

[] Tecnólogo em outro curso. Qual?:_____.

2) Você possui ou está realizando algum curso de especialização?

[] Não.

[] Sim. Qual(is) curso(s)??:_____.

3) Você possui ou está realizando algum curso de mestrado?

[] Não.

[] Sim. Qual(is) curso(s)??:_____.

4) Você possui ou está realizando algum curso de doutorado?

[] Não.

[] Sim. Qual(is) curso(s)??:_____.

5) Você possui alguma certificação?

[] Não.

[] Sim. Qual(is)?

[] PMP.

[] ITIL.

[] MPS.BR (implementador).

[] MPS.BR (avaliador).

[] MCP. Qual(is)?: _____.

[] CBTS.

[] SCP. Qual(is)?: _____.

[] Outra certificação. Qual(is)?: _____.

6) De quais treinamentos você já participou?

[] CMMI

[] Gerência de projetos (PMBOK)

[] Medição de software (GQM e/ou PSM)

[] MPS.BR

[] Pontos de função

[] Qualidade de software

[] Use Case Point

[] Outro relacionado à melhoria de processos ou engenharia de software

Qual(is)?: _____.

7) Você trabalha na área de Computação/Informática?

[] Não.

[] Sim. Em que e quanto tempo?

[] como Técnico de suporte ao usuário. Por ____ anos e ____ meses.

[] como Técnico de manutenção de hardware. Por ____ anos.

[] como Técnico de manutenção de software. Por ____ anos.

[] como Programador. Por ____ anos.

[] como Projetista. Por ____ anos.

[] como Analista. Por ____ anos.

[] como Arquiteto. Por ____ anos.

[] como Testador. Por ____ anos.

[] como Gerente de projetos. Por ____ anos.

[] como Consultor. Por ____ anos.

[] como Instrutor de treinamentos. Por ____ anos.

[] Outro. Qual?:_____ Por ____ anos.

8) Qual o seu conhecimento atual em melhoria de processos?

[] Nenhum

[] Tenho pouco conhecimento do assunto

[] Conheço o básico

[] Aplico na prática com dependência de outros

[] Aplico na prática sem dependência de outros

[] Tenho bastante conhecimento do assunto

[] Sei tudo sobre o assunto

9) Como você considera a área de melhoria de processos?

[] Nada importante

[] Pouco importante

[] Importante

[] Muito importante

10) Você tem interesse em aprender melhoria de processos?

[] Nada interessado

[] Pouco interessado

[] Interessado

[] Muito interessado

APÊNDICE E – PRÉ-TESTE

Neste apêndice é apresentado o pré-teste utilizado nos experimentos para avaliação do jogo SPI City. A Tabela 36 apresenta as questões do pós-teste por nível da taxonomia de Bloom. As respostas corretas estão marcadas em negrito no questionário.

Tabela 36. Questões do pós-teste por nível da taxonomia de Bloom

Questões	Nível da taxonomia de Bloom
Q1, Q3 e Q4	Conhecimento
Q2, Q6, Q7 e Q8	Compreensão
Q5, Q9 e Q10	Aplicação

Pré-Teste

Nome: _____

Grupo: _____

1. Considerando o nível G de maturidade do modelo MPS.BR, qual das atividades abaixo é a **MAIS INDICADA** para iniciar um projeto?

- a) Determinar a equipe do projeto.
- b) Determinar os riscos do projeto.
- c) Determinar o tamanho do projeto.
- d) Elaborar o orçamento do projeto.
- e) **Determinar o escopo do projeto.**

2. Assinale as afirmações corretas sobre ciclo de vida de projeto.

- No nível G não é necessário definir o ciclo de vida que será utilizado nos projetos.
- O ciclo de vida cascata é adequado para projetos de curta duração, pois ele permite maior flexibilidade para alterar os requisitos ao longo do desenvolvimento do projeto.
- O ciclo de vida iterativo é mais adequado para projetos de longa duração, pois permite que mudanças sejam realizadas a cada iteração.**
- São exemplos de fases de ciclo de vida de projeto: iniciação, planejamento, construção e encerramento.**
- O ciclo de vida escolhido deve estar alinhado com as características do projeto.**

3. Considerando o nível G de maturidade do modelo MPS.BR, assinale as atividades que estão relacionadas com a execução do processo de Gerência de Projetos.

Determinar tamanho das tarefas e produtos de trabalho.

Especificar os requisitos.

Determinar esforço das tarefas e produtos de trabalho.

Validar os requisitos.

Determinar custo das tarefas e produtos de trabalho.

Elaborar orçamento.

Elaborar plano de garantia da qualidade.

Elaborar cronograma.

Identificar riscos.

Identificar os artefatos que devem estar sobre controle de versão.

Planejar os recursos humanos e físicos.

Planejar dados.

4. Considerando o nível G de maturidade do modelo MPS.BR, assinale as AFIRMAÇÕES INCORRETAS sobre o processo de Gerência de Requisitos.

Os requisitos podem ser avaliados pela equipe técnica sem a necessidade de critérios objetivos.

Mudanças nos requisitos precisam ser registradas, ter seu impacto avaliado e devem ser aprovadas pelos fornecedores de requisitos.

Os fornecedores de requisitos não precisam ser identificados, uma vez que a Gerência de Requisitos não precisa demonstrar como os requisitos foram levantados (elicitados).

A rastreabilidade pode ser utilizada para apoiar a análise de impacto durante uma solicitação de mudança.

A rastreabilidade entre os requisitos e demais artefatos do projeto não é obrigatória no nível G.

Após a aprovação de uma mudança de requisitos é necessário revisar o plano de projeto e demais artefatos afetados pela mudança.

Modificações no plano de projeto, causadas por uma mudança de requisitos, devem ser revisadas com as pessoas afetadas.

Somente os fornecedores de requisitos podem solicitar mudanças nos requisitos.

5. Uma empresa está se preparando para uma avaliação no nível G de maturidade do MPS.BR e, neste momento, ela está preocupada com as informações que a planilha de identificação de riscos deve conter. Eles querem colocar nesta planilha somente informações que são necessárias para atender o nível G. Marque as informações que você colocaria nesta planilha visando atender somente ao nível G?

Identificação do risco.

Impacto do risco (caso ele ocorra).

Ações de contingência (o que deve ser feito caso o risco ocorra).

Probabilidade de ocorrência do risco.

Fator de exposição (relação entre probabilidade de ocorrência e impacto).

Ações mitigatórias (prevenção).

Prioridade de tratamento do risco.

Gatilhos para que as ações mitigatórias (prevenção) ou corretivas (contingência) sejam disparadas.

Responsável por executar as ações mitigatórias e corretivas.

6. Assinale as afirmações CORRETAS em relação ao resultado esperado “Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.”.

- Dados são todos os atributos do projeto, como por exemplo, esforço, custo, cronograma, requisitos, etc.
- Dados são todos os artefatos que serão gerados ou utilizados durante o projeto.**
- São exemplos de dados do projeto: relatórios, atas de reuniões, documentação, lições aprendidas, e manuais.**
- É necessário especificar as pessoas que poderão ter acesso aos dados, tipo de acesso e forma de acesso.**
- Não é necessário especificar os dados que devem estar sobre controle de versão.**

7. Assinale as afirmações CORRETAS em relação ao seguinte resultado esperado: “GPR2 - As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados”.

- Neste resultado a duração das tarefas e dos produtos de trabalho do projeto deve ser determinada.
- Deve-se utilizar uma técnica, como a Análise de Pontos por Função, para calcular o tamanho das tarefas e dos produtos de trabalhos do projeto.
- O tamanho de um projeto é o número de tarefas do mesmo.
- O tamanho das tarefas e dos produtos de trabalho deve ser determinado utilizando alguma técnica que traga resultados positivos para a organização.**
- O tamanho pode ser utilizado para derivar a estimativa de esforço das tarefas e dos produtos de trabalho.**

8. Assinale a ALTERNATIVA INCORRETA, considerando o que é necessário para implementar o resultado esperado “GPR7 - Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo”.

- a) Deve haver um documento que apresente as habilidades e competências esperadas em cada papel associado ao processo da empresa.
- b) Deve haver algum documento/repositório que guarde os conhecimentos e habilidades de cada colaborador e que permita verificar para quais papéis ele está habilitado.
- c) No momento do planejamento do projeto deve-se determinar o papel responsável por cada atividade do projeto.

d) No momento da alocação dos recursos humanos nas atividades do projeto deve-se analisar o currículo dos colaboradores para verificar quais colaboradores estão habilitados a executarem as atividades de acordo com o papel responsável pela atividade.

e) Quando não houver um colaborador com o perfil desejado para executar uma tarefa, deve-se contratar alguém que possua este perfil.

9. Você é o implementador do nível G de maturidade do MPS.BR numa empresa de desenvolvimento de software. Um dos gerentes está com uma dúvida sobre quais informações ele deve colocar no plano de dados do projeto. Assinale as informações que deveriam constar neste plano.

Critérios para avaliação da qualidade do dado.

Descrição sobre como o dado é acessado.

Descrição do dado.

Histórico de acesso ao dado.

Local onde o dado é armazenado.

Responsável pela criação e manutenção do dado.

Histórico de alterações do dado.

Descrição sobre como o dado é gerado.

Informação se o dado deve estar sobre controle de versão.

Quem tem acesso a este dado e qual o tipo de acesso (leitura/escrita).

10. Você foi contratado por uma empresa para implementar o nível G de maturidade do MPS.BR. Durante uma consultoria, o funcionário responsável pela definição dos processos dessa empresa perguntou que informações ele deveria colocar num template de solicitação de mudança de requisitos. Nesta situação, assinale as informações que deveriam constar neste template.

Nome do projeto na qual a mudança pertence.

Todos os requisitos do projeto.

Descrição da mudança.

Status da mudança (Aprovada / Reprovada).

Árvore de rastreabilidade completa do projeto.

Identificador da mudança.

Nome do solicitante da mudança.

Todos os stakeholders do projeto.

Tipo de mudança (novo, alteração, exclusão de requisito).

Data da solicitação de mudança.

Todos os riscos do projeto.

Impacto da mudança em relação a custo, prazo e esforço.

APÊNDICE F – PÓS-TESTE

Neste apêndice é apresentado o pós-teste utilizado nos experimentos para avaliação do jogo SPI City. A Tabela 37 apresenta as questões do pós-teste por nível da taxonomia de Bloom. As respostas corretas estão marcadas em negrito no questionário.

Tabela 37. Questões do pós-teste por nível da taxonomia de Bloom

Questões	Nível da taxonomia de Bloom
Q2, Q3 e Q4	Conhecimento
Q1, Q5, Q6 e Q8	Compreensão
Q7, Q9 e Q10	Aplicação

Pós-Teste

Nome: _____

Grupo: _____

1. Assinale as afirmações corretas sobre ciclo de vida de projeto.

- O ciclo de vida iterativo é mais adequado para projetos de longa duração, pois permite que mudanças sejam realizadas a cada iteração.**
- O ciclo de vida escolhido deve estar alinhado com as características do projeto.**
- No nível G não é necessário definir o ciclo de vida que será utilizado nos projetos.
- São exemplos de fases de ciclo de vida de projeto: iniciação, planejamento, construção e encerramento.**
- O ciclo de vida cascata é adequado para projetos de curta duração, pois ele permite maior flexibilidade para alterar os requisitos ao longo do desenvolvimento do projeto.

2. Considerando o nível G de maturidade do modelo MPS.BR, qual das atividades abaixo é a MAIS INDICADA para iniciar um projeto?

- a) Elaborar o orçamento do projeto.
- b) Determinar os riscos do projeto.
- c) Determinar a equipe do projeto.
- d) Determinar o escopo do projeto.**
- e) Determinar o tamanho do projeto.

3. Considerando o nível G de maturidade do modelo MPS.BR, assinale as atividades que NÃO estão relacionadas com a execução do processo de Gerência de Projetos.

Determinar tamanho das tarefas e produtos de trabalho.

Especificação dos requisitos.

Determinar esforço das tarefas e produtos de trabalho.

Validação dos requisitos.

Determinar custo das tarefas e produtos de trabalho.

Elaborar orçamento.

Elaboração de plano de garantia da qualidade.

Elaborar cronograma.

Identificar riscos.

Identificação de artefatos que devem estar sob controle de versão.

Planejar os recursos humanos e físicos.

Planejar dados.

4. Considerando o nível G de maturidade do modelo MPS.BR, assinale as AFIRMAÇÕES CORRETAS sobre o processo de Gerência de Requisitos.

Os requisitos podem ser avaliados pela equipe técnica sem a necessidade de critérios objetivos.

Mudanças nos requisitos precisam ser registradas, ter seu impacto avaliado e devem ser aprovadas pelos fornecedores de requisitos.

Os fornecedores de requisitos não precisam ser identificados, uma vez que a Gerência de Requisitos não precisa demonstrar como os requisitos foram levantados (elicitados).

A rastreabilidade pode ser utilizada para apoiar a análise de impacto durante uma solicitação de mudança.

A rastreabilidade entre os requisitos e demais artefatos do projeto não é obrigatória no nível G.

Após a aprovação de uma mudança de requisitos é necessário revisar o plano de projeto e demais artefatos afetados pela mudança.

Modificações no plano de projeto, causadas por uma mudança de requisitos, devem ser revisadas com as pessoas afetadas.

Somente os fornecedores de requisitos podem solicitar mudanças nos requisitos.

5. Assinale as afirmações INCORRETAS em relação ao resultado esperado “Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.”

Dados são todos os atributos do projeto, como por exemplo, esforço, custo, cronograma, requisitos, etc.

Dados são todos os artefatos que serão gerados ou utilizados durante o projeto.

- São exemplos de dados do projeto: relatórios, atas de reuniões, documentação, lições aprendidas, e manuais.
 - É necessário especificar as pessoas que poderão ter acesso aos dados, tipo de acesso e forma de acesso.
 - Não é necessário especificar os dados que devem estar sobre controle de versão.
6. Assinale as afirmações INCORRETAS em relação ao seguinte resultado esperado: “GPR2 - As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados”.
- Neste resultado a duração das tarefas e dos produtos de trabalho do projeto deve ser determinada.**
 - Deve-se utilizar uma técnica, como a Análise de Pontos por Função, para calcular o tamanho das tarefas e dos produtos de trabalhos do projeto.**
 - O tamanho de um projeto é o número de tarefas do mesmo.**
 - O tamanho das tarefas e dos produtos de trabalho deve ser determinado utilizando alguma técnica que traga resultados positivos para a organização.
 - O tamanho pode ser utilizado para derivar a estimativa de esforço das tarefas e dos produtos de trabalho.
7. Uma empresa está se preparando para uma avaliação no nível G de maturidade do MPS.BR e, neste momento, ela está preocupada com as informações que a planilha de identificação de riscos deve conter. Eles querem colocar nesta planilha somente informações que são necessárias para atender o nível G. Marque as informações que você colocaria nesta planilha visando atender somente ao nível G?
- Fator de exposição (relação entre probabilidade de ocorrência e impacto).**
 - Gatilhos para que as ações mitigatórias (prevenção) ou corretivas (contingência) sejam disparadas.
 - Responsável por executar as ações mitigatórias e corretivas.
 - Probabilidade de ocorrência do risco.**
 - Ações corretivas (o que deve ser feito caso o risco ocorra).
 - Ações mitigatórias (prevenção).
 - Prioridade de tratamento do risco.**
 - Impacto do risco (caso ele ocorra).**
 - Identificação do risco.**
8. Assinale a ALTERNATIVA INCORRETA, considerando o que é necessário para implementar o resultado esperado “GPR7 - Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo”.
- a) No momento da alocação dos recursos humanos nas atividades do projeto deve-se analisar o currículo dos colaboradores para verificar quais colaboradores estão habilitados a executarem as atividades de acordo com o papel responsável pela atividade.

- b) Quando não houver um colaborador com o perfil desejado para executar uma tarefa, deve-se contratar alguém que possua este perfil.
- c) Deve haver um documento que apresente as habilidades e competências esperadas em cada papel associado ao processo da empresa.
- d) No momento do planejamento do projeto deve-se determinar o papel responsável por cada atividade do projeto.
- e) Deve haver algum documento/repositório que guarde os conhecimentos e habilidades de cada colaborador e que permita verificar para quais papéis ele está habilitado.
9. Você foi contratado por uma empresa para implementar o nível G de maturidade do MPS.BR. Durante uma consultoria, o funcionário responsável pela definição dos processos dessa empresa perguntou que informações ele deveria colocar num template de solicitação de mudança de requisitos. Nesta situação, assinale as informações que deveriam constar neste template.
- Identificador da mudança.**
 - Nome do solicitante da mudança.**
 - Todos os requisitos do projeto.
 - Tipo de mudança (novo, alteração, exclusão de requisito).**
 - Todos os stakeholders do projeto.
 - Impacto da mudança em relação a custo, prazo e esforço.**
 - Todos os riscos do projeto.
 - Data da solicitação de mudança.**
 - Descrição da mudança.**
 - Status da mudança (Aprovada / Reprovada).**
 - Árvore de rastreabilidade completa do projeto.
 - Nome do projeto na qual a mudança pertence.**
10. Você é o implementador do nível G de maturidade do MPS.BR numa empresa de desenvolvimento de software. Um dos gerentes está com uma dúvida sobre quais informações ele deve colocar no plano de dados do projeto. Assinale as informações que deveriam constar neste plano.
- Descrição do dado.**
 - Local onde o dado é armazenado.**
 - Descrição sobre como o dado é gerado.**
 - Critérios para avaliação da qualidade do dado.
 - Descrição sobre como o dado é acessado.**
 - Histórico de acesso ao dado.
 - Responsável pela criação e manutenção do dado.**
 - Histórico de alterações do dado.
 - Quem tem acesso a este dado e qual o tipo de acesso (leitura/escrita).**
 - Informação se o dado deve estar sobre controle de versão.

APÊNDICE G – QUESTIONÁRIO QUALITATIVO

Este apêndice apresenta o questionário utilizado nas avaliações do jogo SPI City para avaliar a percepção dos participantes sobre o jogo. Este questionário foi retirado do modelo de avaliações de jogos educacionais (SAVI, 2012). Para a avaliação do SPI City foram adicionadas algumas questões sobre aprendizagem de melhoria de processo de software (três últimas).

Questionário de avaliação de jogos educacionais

<Nome do Jogo> - <Data da avaliação>

Gostaríamos que você respondesse as questões abaixo para nos ajudar a melhorar este jogo. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa. Algumas fotografias poderão ser feitas como registro desta atividade, mas não serão publicadas em nenhum local sem autorização.

<Nome do Pesquisador> - <E-mail>

<Local>

Disciplina e turma: _____

Por favor, circule **um número** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

Afirmações	Sua avaliação			Comentários sobre a questão
O design do jogo é atraente (interface ou objetos, como cartas ou tabuleiros).	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Houve algo interessante no início do jogo que capturou minha atenção.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
A variação (de forma, conteúdo ou de atividades) ajudou a me manter atento ao jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O conteúdo do jogo é relevante para os meus interesses.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O funcionamento deste jogo está adequado ao meu jeito de aprender.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O conteúdo do jogo está conectado com outros conhecimentos que eu já possuía.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Foi fácil entender o jogo e começar a utilizá-lo como material de estudo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Ao passar pelas etapas do jogo senti confiança de que estava aprendendo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Estou satisfeito porque sei que terei oportunidades de utilizar na prática	Discordo	-2-1 0 +1 +2	Concordo	

coisas que aprendi com o jogo.	Fortemente		Fortemente	
É por causa do meu esforço pessoal que consigo avançar no jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Temporariamente esqueci das minhas preocupações do dia-a-dia, fiquei totalmente concentrado no jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Eu não percebi o tempo passar enquanto jogava, quando vi o jogo acabou.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me senti mais no ambiente do jogo do que no mundo real, esquecendo do que estava ao meu redor.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Pude interagir com outras pessoas durante o jogo	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me diverti junto com outras pessoas	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo promove momentos de cooperação e/ou competição entre as pessoas que participam.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Este jogo é adequadamente desafiador para mim, as tarefas não são muito fáceis nem muito difíceis.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo evolui num ritmo adequado e não fica monótono – oferece novos obstáculos, situações ou variações de atividades.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me diverti com o jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Quando interrompido, fiquei desapontado que o jogo tinha acabado (gostaria de jogar mais).	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Eu recomendaria este jogo para meus colegas.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Gostaria de utilizar este jogo novamente	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Consegui atingir os objetivos do jogo por meio das minhas habilidades.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Tive sentimentos positivos de eficiência no desenrolar do jogo	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo contribuiu para a minha aprendizagem na disciplina	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
A experiência com o jogo vai contribuir para meu desempenho na vida profissional.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo ajuda a LEMBRAR os resultados esperados do MPS.BR.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo ajuda a ENTENDER os resultados esperados do MPS.BR.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo ajuda a APLICAR os resultados esperados do MPS.BR.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	

– Cite 3 pontos fortes do

jogo: _____

– Por favor, dê 3 sugestões para a melhoria do

jogo: _____
