

MATHEUS MOREIRA DE CARVALHO

**MaintES - UM JOGO EDUCACIONAL PARA APOIAR
O PROCESSO DE ENSINO-APRENDIZAGEM
SOBRE MANUTENÇÃO DE SOFTWARE**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS - BRASIL
2014

MATHEUS MOREIRA DE CARVALHO

**MaintES - UM JOGO EDUCACIONAL PARA APOIAR
O PROCESSO DE ENSINO-APRENDIZAGEM
SOBRE MANUTENÇÃO DE SOFTWARE**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:
Engenharia de Software

Orientador:
Prof. Dr. Heitor Augustus Xavier Costa

LAVRAS
MINAS GERAIS - BRASIL
2014

**Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca
Central da UFLA**

Carvalho, Matheus Moreira de

MaintES - Um Jogo Educacional para Apoiar o Processo de Ensino-Aprendizagem sobre Manutenção de Software / Matheus Moreira de Carvalho. Lavras - Minas Gerais, 2014. XXX p.

Monografia de Graduação - Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Jogo Educacional. 2. Manutenção de Software. 3. Informática na Educação. I. Carvalho, M. M. de II. Universidade Federal de Lavras. III. MaintES - Um Jogo Educacional para Apoiar o Processo de Ensino-Aprendizagem sobre Manutenção de Software.

MATHEUS MOREIRA DE CARVALHO

**MAINTES - UM JOGO EDUCACIONAL PARA
APOIAR O PROCESSO DE ENSINO-
APRENDIZAGEM SOBRE MANUTENÇÃO DE
SOFTWARE**

Monografia de graduação apresentada ao
Colegiado do Curso de Bacharelado em
Ciência da Computação, para obtenção
do título de Bacharel.

APROVADA em 11 de julho de 2014.

José Monserrat Neto

Rêmulo Maia Alves



Heitor Augustus Xavier Costa (Orientador)

LAVRAS-MG
2014

AGRADECIMENTOS

Agradeço à minha mãe e ao meu pai por todo apoio que me deram nessa etapa da minha vida.

À minha namorada Thamiris Karlovic pelo apoio, incentivo, companhia, amizade e força, fundamentais na obtenção desta conquista.

Ao professor Dr. Heitor Augustus Xavier Costa pela excelente orientação, sempre bem disposto, dedicação, amizade e conhecimento transmitido no desenvolvimento deste trabalho.

Aos amigos de graduação Bruno, Sérgio e Paulo Roberto, pelo companheirismo e camaradagem.

À Universidade Federal de Lavras (UFLA) pela oportunidade de realizar o curso de Graduação e aos professores do Departamento de Ciência da Computação da UFLA pelos conhecimentos e experiências transmitidos.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1. INTRODUÇÃO	1
1.1. Motivação.....	2
1.2. Objetivo.....	2
1.3. Metodologia de Desenvolvimento	3
1.3.1. Procedimentos Metodológicos	3
1.4. Estrutura do Trabalho	4
2. JOGOS EDUCACIONAIS	5
2.1. Considerações Iniciais	5
2.2. Características	5
2.3. Classificação dos Jogos	5
2.4. Vantagens do Uso de Jogos na Educação	6
2.5. Jogos Computacionais	8
2.6. Considerações Finais	9
3. MANUTENÇÃO DE SOFTWARE	10
3.1. Considerações Iniciais	10
3.2. Manutenibilidade	10
3.3. Processo de Manutenção de Software	12
3.3.1. ISO/IEC 12207	12
3.3.2. ISO/IEC 14764	15
3.3.3. Melhorias no Processo de Software.....	16
3.3.3.1. <i>Software Maintenance Maturity Model</i>	16
3.3.3.2. <i>Corrective Maintenance Maturity Model</i>	17
3.4. Considerações Finais	18
4. TRABALHOS RELACIONADOS	19
5. MaintES - <i>MAINTENANCE EDUCATIONAL SOFTWARE</i>	22
5.1. Considerações Iniciais	22
5.2. Especificações do Software.....	22
5.3. Modelagem.....	22
5.3.1. Diagrama de Casos de Uso.....	22
5.3.2. Modelo Relacional	23
5.4. Funcionamento	23
5.4.1. Organização Modular	25
5.4.2. Módulo Administrador	25
5.4.3. Módulo Jogador	31

5.5. Considerações Finais	40
6. AVALIAÇÃO	42
6.1. Considerações Iniciais	42
6.2. Resultados	42
6.3. Considerações Finais	46
7. CONSIDERAÇÕES FINAIS.....	48
7.1. Conclusões.....	48
7.2. Contribuições.....	48
7.3. Limitações	48
7.4. Trabalhos Futuros	49
REFERÊNCIAS BIBLIOGRÁFICAS	50

LISTA DE FIGURAS

Figura 3-1 - Grupos de Processos de Ciclo de Vida de Software (Fonte: [ISO/IEC, 2008]).....	13
Figura 3-2 - Atividades de Manutenção de Software (Fonte: [Paduelli, 2007])	15
Figura 3-3 - Processos de Manutenção de Software (Fonte: [ISO/IEC, 2006])	16
Figura 5-1 - Diagrama de Casos de Uso.....	23
Figura 5-2 - Modelo Relacional	24
Figura 5-3 - Tela de Entrada do Jogo - Autenticação	26
Figura 5-4 - Tela Inicial do Administrador	26
Figura 5-5 - Tela de Alteração de Dados Cadastrais de Jogadores	27
Figura 5-6 - Tela de Alteração de Dados Cadastrais de Administrador	27
Figura 5-7 - Tela de Gerenciamento de Perguntas	28
Figura 5-8 - Tela de Configurações Gerais.....	29
Figura 5-9 - Tela de Configurações dos Treinamentos	30
Figura 5-10 - Tela de Configurações dos Torneios	30
Figura 5-11 - Tela Inicial do Jogador	31
Figura 5-12 - Ambiente do MaintES	32
Figura 5-13 - Regras do MaintES.....	32
Figura 5-14 - Tela para Atualizar Dados Cadastrais do Jogador	33
Figura 5-15 - <i>Rank</i>	33
Figura 5-16 - Tela de Estudo.....	34
Figura 5-17 - Textos Resumos	35
Figura 5-18 - Treinamento	36
Figura 5-19 - Tela de Conquistas	37
Figura 5-20 - Tela de Gerenciamento da Equipe	38
Figura 5-21 - Entrada no Torneio	38
Figura 5-22 - Torneio Iniciante e Janela da Equipe	39
Figura 5-23 - Tela Apresentada ao Jogador que Venceu o Jogo	40
Figura 5-24 - Tela Apresentada ao Jogador que Perdeu o Jogo	40
Figura 6-1 - Idade dos Avaliadores	43

Figura 6-2 - Sexo dos Avaliadores	43
Figura 6-3 - Escolaridade dos Avaliadores.....	43
Figura 6-4 - Conhecimento dos Avaliadores	44
Figura 6-5 - Experiência dos Avaliadores	44
Figura 6-6 - Ludicidade do MaintES	45
Figura 6-7 - Sentimento do Usuário em Relação ao MaintES	45
Figura 6-8 - Avaliação sobre o Conteúdo de Manutenção de Software.....	46

LISTA DE TABELAS

Tabela 2-1 - Vantagens dos Jogos (Fonte: [Grando, 2001])	7
Tabela 2-2 - Desvantagens dos Jogos (Fonte: [Grando, 2001])	7
Tabela 5-1 - Conquistas e Recompensas	37

MaintES - UM JOGO EDUCACIONAL PARA APOIAR O PROCESSO DE ENSINO-APRENDIZAGEM SOBRE MANUTENÇÃO DE SOFTWARE

RESUMO

Atualmente, há alternativas para melhorar o processo de ensino-aprendizagem em várias áreas de estudo. Embora isso seja um importante avanço, a Engenharia de Software ainda apresenta déficit na quantidade de opções disponíveis para auxiliar os estudantes que procuram adquirir conhecimento. Neste trabalho, é apresentado o desenvolvimento do MaintES (*Maintenance Educational Software*), um jogo educacional computacional 3D para envolver o estudante em um ambiente interativo repleto de desafios relacionados a Manutenção de Software. Este software é disponibilizado pela plataforma Web e o estudante necessita apenas de um computador com acesso a Internet para jogar. Para avaliar a sua utilidade na contribuição no processo de ensino-aprendizagem, foram convidados alunos, professores e pesquisadores para avaliarem o MaintES. Após eles jogarem o MaintES, eles responderam um questionário no qual havia questões para retratar a ludicidade, o sentimento do usuário e o conteúdo sobre manutenção de software. Com a análise das respostas, foi possível verificar que o MaintES pode contribuir significativamente para melhorar a aprendizagem do conteúdo de Manutenção de Software.

Palavras-chave: Manutenção de software, Jogos Educacionais, Engenharia de Software

MaintES - AN EDUCATIONAL GAME TO SUPPORT TEACHING-LEARNING PROCESS ON SOFTWARE MAINTENANCE

ABSTRACT

There are alternatives to improve teaching-learning process in various fields of study. Although these alternatives are important advancement, Software Engineering still presents deficit in number of available options to assist students who want to acquire knowledge about this area. This work presents MaintES (*Maintenance Educational Software*), 3D educational software that seeks to involve students a challenger and interactive environment related to Software Maintenance. This software is provided by the Web platform and the student needs only a computer with Internet access to play. To evaluate its usefulness in contributing in the teaching-learning process, we invited students, teachers, and researchers to assess MaintES. After they play MaintES, they answered a questionnaire in which questions had to know playfulness, user feeling, and content on software maintenance. With the analysis of the responses, we found that MaintES can contribute significantly to better learning of the contents of Software Maintenance.

Keywords: Software Maintenance, Educational Games, Software Engineering.

1. INTRODUÇÃO

Na sociedade atual, cada vez mais se exige que os profissionais tenham formação qualificada, melhorando, evoluindo e aperfeiçoando sua capacitação profissional. Uma das preocupações das Instituições de Ensino Superior é formar profissionais capacitados e aptos para a realização de suas funções [Enricone, 2008]. A realização de trabalhos práticos, para os alunos terem melhor assimilação do conhecimento passado pelos docentes nas disciplinas da área Computação e Informática e estejam bem capacitados a aplicá-los, é um consenso geral entre os professores que lecionam disciplinas pertinentes a essa área de conhecimento [Schneider; Johnston, 2003]. Um fator negativo na utilização de aulas expositivas é estar ativo apenas o sentido da visão e da audição. Uma alternativa é o uso de eventos simulados e atividades vivências, que permitem assimilar diversas situações de forma prática e interativa. O envolvimento de um indivíduo com determinada atividade prática resulta em melhores resultados, pois eles tornam-se responsáveis pelas tomadas de decisões, ações, escolhas e pelo rumo que a atividade irá seguir [Prikladnicki *et al.*, 2009].

Na Engenharia de Software, não é diferente. Prender a atenção do aluno e proporcionar vontade de aprender tendem a ser maneiras eficazes de aprimorar o processo de ensino-aprendizagem. Em confronto a essa ideia, há as massivas aulas teóricas de Engenharia de Software em que uma expressiva quantidade de informações é passada aos alunos de forma exclusivamente expositiva, sem aplicação prática, o que causa aos alunos desmotivação e dificuldade na fixação do conteúdo apresentado. Com isso, o processo de ensino-aprendizagem na área Engenharia de Software, em especial o assunto Manutenção de Software, sofre por utilizar esses padrões tradicionais de ensino. Dessa forma, uma alternativa é a utilização de Jogos Computacionais voltados para a educação, cujo objetivo é auxiliar na aprendizagem dos discentes, utilizando fundamentos como regras, metas/objetivos, estratégias e competição. Com os jogos, o aluno tem contato com ambiente, conteúdo e informações e pode interagir com o jogo proporcionando novas experiências válidas e pertinentes ao assunto abordado.

Contudo, ainda que buscas por soluções, esforços em pesquisas e empenho em desenvolvimentos de jogos computacionais educacionais sejam consideráveis, a maioria desses jogos não possuem o objetivo de apoiar o ensino em Engenharia de Software [Prensky, 2001].

1.1. Motivação

Existem iniciativas nas Instituições de Ensino Superior em adotar a disciplina exclusiva para o ensino do assunto Manutenção de Software em seus cursos de graduação na área de Computação. Isso releva a importância dos profissionais aprenderem o conteúdo sobre esse assunto na sua formação. Porém, em muitas dessas instituições, as aulas são lecionadas aplicando atividades unicamente teóricas [Paduelli, 2007]. Isso pode gerar desconforto ao aprender a disciplina, pois se torna massiva e cansativa; uma alternativa para esse desconforto é a utilização de jogos que abordassem o assunto. Uma vez que os jogos são atrativos e conseguem fixar o jogador, eles tornam-se poderosas ferramentas educacionais, quando associadas às atividades didáticas. O processo de ensino-aprendizagem de Engenharia de Software precisa alcançar os estudantes para que, quando contratados por empresas ou organizações, estejam preparados para enfrentar dificuldades e desafios, incluindo ambientes competitivos e interdisciplinares [Figueiredo *et al.*, 2007].

Dessa forma, a realização deste trabalho foi motivada, principalmente, pelo fato da transferência do conhecimento na área Engenharia de Software, especificamente o assunto Manutenção de Software, tem se dado por ministração de aulas exclusivamente expositivas. Essas aulas são cansativas para alunos e para professores e podem não promover total aprendizado do conteúdo. Isso pode ser creditado a muitos fatores, um deles é a apresentação de métodos, técnicas, processos, tecnologias, metodologias, paradigmas, procedimentos, entre outros conceitos sem aplicações práticas (nas empresas), o que dificulta sua fixação. Portanto, o processo de ensino-aprendizagem de Engenharia de Software precisa ter alternativas válidas, visto a importância dos profissionais terem conhecimento e experiência necessários para realizar suas funções e não ficar limitado apenas às aulas exclusivamente expositivas e teóricas. Uma alternativa para fixar esses conceitos é disponibilizar aos alunos jogos computacionais educacionais (sistemas de software educacionais) para treinamento abordando, por exemplo, o assunto Manutenção de Software.

1.2. Objetivo

Neste trabalho, o objetivo é estudar, desenvolver e apresentar um software educacional (jogo) dedicado ao apoio do processo de ensino-aprendizagem do assunto Manutenção de Software. Sua missão é reduzir impactos negativos trazidos pelas formas

tradicionais de ensino dessa área. Espera-se que esse software possa auxiliar os estudantes a assimilar, consolidar e fixar informações pertinentes ao assunto abordado. Algumas metas foram alcançadas:

- Estudar, aprender e saber utilizar a ferramenta de desenvolvimento de jogos Unity3D;
- Elucidar os benefícios que os jogos trazem para o processo de ensino-aprendizagem;
- Investigar conceitos, técnicas, processos e conhecimentos sobre o assunto Manutenção de Software.

1.3. Metodologia de Desenvolvimento

Este trabalho pode ser classificado como **pesquisa aplicada**, uma vez que o software educacional computacional MaintES foi desenvolvido e pode ser utilizado para apoiar o processo de ensino-aprendizagem. Em relação aos objetivos, este trabalho pode ser classificado como **pesquisa descritiva**, pois o usuário do MaintES aprende como acontecem as atividades de Manutenção de Software. Quanto à abordagem, este trabalho pode ser classificado como **pesquisa qualitativa**, tendo em vista que há busca de aperfeiçoamento do processo de ensino-aprendizagem de assuntos abordados em Manutenção de Software. Os procedimentos envolvidos neste trabalho são classificados como **estudo de caso**, pois foi realizada uma investigação empírica com o jogo desenvolvido, cujo método foi a coleta de dados por meio da aplicação de um **questionário** de avaliação.

1.3.1. Procedimentos Metodológicos

O estudo dos métodos tradicionais utilizados no processo de ensino-aprendizagem de Engenharia de Software, mais especificamente em Manutenção de Software, possibilitou identificar dificuldades encontradas pelos estudantes. Dessa forma, o desenvolvimento de um jogo educacional computacional (MaintES) foi impulsionado pelo desafio de desenvolver uma solução para essas dificuldades. Ainda, a busca por informações referentes à formação dos alunos contribuiu para identificar as necessidades dos futuros profissionais (estudantes e profissionais da área de Computação que atuam em Engenharia de Software/Manutenção de Software).

O embasamento teórico foi adquirido por meio da leitura de artigos científicos e livros relacionados ao assunto. Ainda, reuniões regulares ocorreram entre orientador e orientado para discussão de ideias, busca por artigos científicos pertinentes ao assunto e

investigação de jogos existentes sobre Manutenção de Software. A *engine* Unity3D foi escolhida como ferramenta de desenvolvimento por ser uma *engine* "robusta", permitir a integração entre linguagens de alto nível e banco de dados e contar com diversas funções para facilitar o desenvolvimento de jogos. A escolha foi feita depois de realizar pesquisa sobre ferramentas de desenvolvimento e concluir que a Unity3D poderia melhor atender as características do MaintES.

O MaintES foi disponibilizado na Internet via um servidor de aplicações para que potenciais usuários pudessem utilizá-lo. Em seguida, foram convidadas pessoas especialistas em Engenharia de Software (Manutenção de Software) e alunos em formação de algumas Instituições de Ensino Superior para serem avaliadores do MaintES. Para isso, foram criados e disponibilizados *login* e senha para essas pessoas autenticarem-se e jogarem. Além disso, um questionário de avaliação foi elaborado e disponibilizado a essas pessoas para entender a percepção delas quando utilizaram o MaintES.

1.4. Estrutura do Trabalho

O restante deste trabalho está organizado da seguinte forma.

Características e vantagens sobre a utilização de jogos na educação, inclusive de jogos computacionais, e classificação dos diferentes tipos de jogos são descritas no Capítulo 2.

Assuntos técnicos, incluindo métodos, práticas e normas, sobre Manutenção de Software são relatados no Capítulo 3.

Trabalhos relacionados com o trabalho atual, encontrados durante pesquisas de revisão bibliográfica, são descritos de forma resumida no Capítulo 4.

Funcionamento, dinâmica e modelagem do jogo são descritos no Capítulo 5.

Avaliação do jogo é apresentada no Capítulo 6.

Conclusões, contribuições, limitações e sugestões de trabalhos futuros são discutidas no Capítulo 7.

2. JOGOS EDUCACIONAIS

2.1. Considerações Iniciais

Com a crescente exigência do mercado por profissionais capacitados e preparados, há preocupação sobre os métodos e as técnicas de aprendizagem para garantir a excelência desses profissionais. Em razão disso, a utilização de jogos na educação mostra-se interessante e promissora, uma vez que possuem potencial para auxiliar os processos de aprendizagem [Silva, 2008].

Jogos educacionais são caracterizados na Seção 3.2. Tipos de jogos existentes são apresentados na Seção 3.3. Vantagens de utilizar esses jogos na educação são discutidas na Seção 3.4. Importância de jogos computacionais no processo de ensino-aprendizagem é abordada na Seção 3.5.

2.2. Características

Com a utilização de jogos educativos em ambientes escolares, os professores podem estimular seus alunos a aprenderem o conteúdo passado. Isso auxilia, otimiza e aperfeiçoa o trabalho do professor para transmitir o conteúdo, pois, uma vez que os alunos estão em contato com o objeto de estudo, o professor pode dedicar-se a tirar dúvidas individualmente. Por outro lado, a dinâmica do jogo proporciona aos estudantes a compreensão de regras, a fixação do objeto de estudo e o desenvolvimento de habilidades e aspectos sociais entre os alunos [Feitosa; Campos, 2010]. Os jogos possuem características fundamentais [Huizinga, 1971 *apud* Moratori, 2003]: i) ser dinâmico; ii) exigir que os jogadores observem e respeitem as regras do jogo; iii) permitir a análise e a comparação dos resultados, por ser possível repetir o jogo quantas vezes forem necessárias; e iv) ser uma atividade que permita a livre ação do jogador, mas deve existir limites de tempo e de espaço para evitar a fuga do objetivo proposto pelo jogo.

2.3. Classificação dos Jogos

A classificação dos jogos é feita de acordo com os seus objetivos que podem ser encontrados em jogos computacionais e em jogos tradicionais (tabuleiros, cartas, etc.) [Oliveira, 2013]. Dentre os tipos de jogos, destacam-se [Tarouco *et al.*, 2004]:

- **Ação.** Nesses jogos, há influência direta no desenvolvimento de habilidades psicomotoras dos jogadores, por exemplo, reflexo, coordenação e agilidade na tomada

de decisões. O ideal é o jogo apresentar momentos intelectuais intercalados com momentos de atividade motora. Dessa forma, o poder de raciocínio é aprimorado, uma vez que o jogador precisa responder rapidamente aos estímulos do jogo;

- **Aventura.** Nesses jogos, os jogadores possuem domínio sobre o ambiente em que farão parte. Quando bem projetados e bem modelados, esses jogos permitem aos jogadores vivenciarem situações que dificilmente vivenciaríamos em um ambiente escolar, por exemplo, a condição real de trabalho em uma empresa;
- **Estratégia.** Nesses jogos, a principal característica é estimular o raciocínio e o planejamento. Os jogadores são obrigados a administrar, a construir e a planejar para concluir o objetivo do jogo;
- **Role-playing game (RPG).** Nesses jogos, os jogadores controlam personagens livres para interagir com o ambiente e com outros usuários. Com isso, a evolução do jogo ocorre conforme a vontade e o interesse do jogador, uma vez que o jogo não interfere em suas ações no jogo;
- **Lógica.** Nesses jogos, os jogadores são mais desafiados. Embora estimulem o raciocínio lógico, eles não trabalham as habilidades motoras. Esse problema pode ser contornado com a adição de temporizadores para exigir mais agilidade e reflexos.

Em ambientes escolares, a utilização de jogos computacionais educacionais motiva os estudantes e ajuda no desenvolvimento de alguns aspectos, tais como, concentração e persistência na realização de tarefas. Esses jogos são responsáveis pela flexibilidade cognitiva, pois funcionam como ginástica mental, aumentando a quantidade de conexões neurais [Tarouco *et al.*, 2004]. A aprendizagem do aluno por meio desses jogos acontece de maneira autônoma, com a interação entre o jogador e o ambiente, tomada de decisões e descobertas realizadas durante o jogo. Com isso, a utilização desses jogos pode ampliar o desenvolvimento de competências e habilidades individuais, o que permite o surgimento de uma forma diferente de relação entre professores e alunos.

2.4. Vantagens do Uso de Jogos na Educação

O uso de jogos como uma ferramenta de apoio ao processo de ensino-aprendizagem apresenta vantagens, tais como, concentração, cooperação, motivação, competição e interação [Borges, 2005]. A utilização desses jogos na educação, principalmente aqueles que proporcionam aos jogadores a simulação de situações e ambientes, tem gerado

resultados positivos em termos de motivação e de concentração dos estudantes, mesmo quando empregado em diversas áreas de conhecimento [Catapan *et al.*, 1999].

O envolvimento do indivíduo com o objeto de estudo é primordial ao seu desenvolvimento cognitivo. A utilização de jogos no processo de ensino-aprendizagem é uma alternativa importante para o seu envolvimento prático com o objeto de estudo, em que a motivação é o elemento responsável por impulsionar esse processo. Com isso, situações e perspectivas impostas pelo jogo têm o objetivo de provocar e estimular o indivíduo. Essa provocação ocorre no sentido de fazer pensar ou refletir, pois o jogador é impulsionado a rever as ações tomadas, estimulando sua estrutura cognitiva [Vasconcellos, 2004].

Jogar permite aos alunos entenderem as regras, terem-nas bem definidas e fazerem com que identifiquem em quais contextos elas devem ser aplicadas, modificando-as quando houver novos contextos. No decorrer do jogo, fatores como autonomia, criatividade, situações adversas e poder criativo são revelados [Tarouco *et al.*, 2004]. A inserção de jogos no contexto do processo de ensino-aprendizagem implica em vantagens (Tabela 2-1) e desvantagens (Tabela 2-2) [Grando, 2001]. Por exemplo, o jogo requer a participação ativa do aluno na construção do seu conhecimento e a utilização dos jogos como fator de motivacional aos alunos são duas vantagens; por outro lado, a constante interferência do professor faz com que o jogo perca a ludicidade e a coerção do professor, exigindo que o aluno jogue são duas desvantagens.

Tabela 2-1 - Vantagens dos Jogos (Fonte: [Grando, 2001])

Vantagens
Fixar conceitos aprendidos de forma motivadora ao aluno.
Introduzir e desenvolver conceitos de difícil compreensão.
Desenvolver estratégias de resolução de problemas (desafio dos jogos).
Capacitar em tomar decisões e saber avaliá-las.
Por significado para conceitos aparentemente incompreensíveis.
Propiciar o relacionamento de diferentes disciplinas (interdisciplinaridade).
Requer a participação ativa do aluno na construção do seu próprio conhecimento.
Favorecer a socialização entre alunos e a conscientização do trabalho em equipe.
Utilizar jogos é fator de motivação para os alunos.
Favorecer o desenvolvimento da criatividade, de senso crítico, da participação, da competição sadia, da observação, das várias formas de uso da linguagem e do resgate do prazer em aprender.
Utilizar as atividades com jogos para reforçar/recuperar habilidades de que os alunos necessitem, sendo útil no trabalho com alunos de diferentes níveis.
Permitir ao professor identificar e diagnosticar erros de aprendizagem e de atitudes e dificuldades dos alunos.

Tabela 2-2 - Desvantagens dos Jogos (Fonte: [Grando, 2001])

Desvantagens
Dar ao jogo caráter puramente aleatório, tornando-se "apêndice" em sala de aula, quando mal utilizados. Os alunos jogam e sentem-se motivados apenas pelo jogo, sem saber o porquê jogam.

Tabela 3-2 - Desvantagens dos Jogos (Fonte: [Grando, 2001]) (cont.)

Desvantagens
Gastar mais tempo com as atividades de jogo em sala de aula. Se o professor não estiver preparado, pode existir "sacrifício" de outros conteúdos pela falta de tempo.
Ter falsas concepções de que devem ensinar os conceitos utilizando jogos. Em geral, as aulas transformam-se em verdadeiros cassinos, sem sentido algum para o aluno.
Perder a "ludicidade" do jogo pela interferência constante do professor, destruindo a essência do jogo.
Existir a coerção do professor, exigindo que o aluno jogue, mesmo que ele não queira, destruindo a voluntariedade pertencente à natureza do jogo.
Ter dificuldade de acesso e de disponibilidade de materiais e de recursos sobre a utilização de jogos no ensino que possam vir a subsidiar o trabalho docente.

2.5. Jogos Computacionais

O computador está presente na realização de tarefas cotidianas das pessoas. Sobre a ótica educacional, há discussão a respeito do uso do computador como um instrumento no processo de ensino-aprendizagem, havendo questões que precisam ser respondidas. O computador é um dos principais motivadores para a continuidade do avanço tecnológico, não pode mais ser ignorado pela escola, mas o desafio é fazer com que seja utilizado no processo de educação de forma efetiva e eficiente. Ele deve ser aliado a projetos escolares com o preparo de futuros cidadãos [Smole; Diniz, 2001]. O processo de ensino-aprendizagem passa por um processo de revolução por causa da introdução do computador no âmbito escolar [Valente, 1993]. O uso da tecnologia em atividades educacionais tem se tornado comum, principalmente no ensino básico [Krüger *et al.*, 2001]. Para efetiva implantação desse uso, são necessários quatro "ingredientes" [Valente, 1993]: i) computador; ii) software educativo; iii) professor capacitado para utilizar o computador como meio educacional; e iv) aluno.

É ampla a área de atuação do computador, pois exerce influência (in)direta nas camadas sociais e nas atividades humanas. Na educação, o computador tem sido utilizado para ensinar sobre computação (ensino de computação) e ensinar qualquer assunto (ensino utilizando computador) [Valente, 1993]. Um jogo computacional educacional deve ser avaliado para descobrir se atende ou não as necessidades e as propostas de ensino adotadas [Gomes *et al.*, 2002]. Sendo o professor o "condutor" do ensino, cabe a ele efetuar a escolha e o uso desse jogo. Ele deve definir quais parâmetros de qualidade devem ser adequados às necessidades dos alunos [Gomes *et al.*, 2002]. Se o professor falha nessa escolha, o processo de ensino-aprendizagem pode ser comprometido. Em consequência dessa falha e de outros fatores (por exemplo, sistema, funcionamento e estrutura física da escola), a prática da informática na escola, muitas vezes, distancia-se de seu caráter pedagógico. Por um lado, o computador "ensina" o aluno com o software; por outro lado, o

aluno "ensina" o computador com o software, o que torna o ensino uma "via de mão dupla". Deve-se atentar quanto aos reais objetivos na utilização do computador na educação, pois sua introdução na educação não deve ser por causa do modismo ou de inovação tecnológica [Valente, 1993].

Os jogos computacionais educacionais são excelentes ferramentas para instruir os alunos, a ideia de motivar enquanto se diverte é uma combinação interessante para estimulá-los e aumentar sua capacidade de retenção do que foi ensinado, pois esses jogos passam a exercitar mais funções cognitivas e intelectuais do jogador [Tarouco *et al.*, 2004]. O jogo computacional educacional não substitui outras fontes de consulta e não supre lacunas que o professor pode deixar. Assim, ele deve ser utilizado como ferramenta complementar no processo de ensino-aprendizagem, mas sua dinâmica provê ao aluno mais agilidade na busca de informações que sirvam ao seu aprendizado [Lopes *et al.*, 2004]. Esses jogos são apontados como um promissor material de ensino, que pode trazer benefícios para a educação. Os pesquisadores devem empenhar mais esforços para constatar/evidenciar as vantagens pedagógicas que o computador possa oferecer para a aprendizagem dos alunos [Wangenheim *et al.*, 2009].

2.6. Considerações Finais

O avanço e o desenvolvimento contínuo de novas tecnologias ajudam na criação e na disponibilização de novas alternativas que apoiem a educação e o ensino. Isso pode deixar os métodos tradicionais de ensino em um estado obsoleto, bem como pensamentos e técnicas relacionados a esses métodos.

Desse modo, os jogos educacionais fazem parte dessas novas alternativas disponíveis e podem colaborar significativamente na educação, principalmente no processo de ensino-aprendizagem. As características que os jogos proporcionam aos jogadores despertam o interesse e a motivação. Assim, o processo de ensino-aprendizagem passa a contar com ferramentas poderosas, que auxiliam e tornam esse processo mais eficiente.

3. MANUTENÇÃO DE SOFTWARE

3.1. Considerações Iniciais

A atividade de manutenção de software é dificultada, principalmente, pela documentação inexistente, incompleta e/ou desatualizada. Dessa forma, para facilitar o entendimento das funções do software legado, é preciso realizar a recuperação dessa documentação, o que eleva os custos dessa atividade [Pressman, 2009; Sommerville, 2010]. Sistemas de software existentes necessitam de manutenção, seja para correção de falhas e/ou melhorias. Desse modo, há evolução de estudos e de técnicas relacionadas à atividade de manutenção de software.

Atividades relacionadas à Manutenção de Software são detalhadas na Seção 4.2. Processos de manutenção de software são abordados na Seção 4.3.

3.2. Manutenibilidade

O processo de desenvolvimento de sistemas de software está completo quando ele foi entregue e está sendo utilizado pelos usuários. Qualquer mudança, após o produto de software estar operacional, é considerada manutenção. Como mudanças são inevitáveis ao longo da sua vida, mecanismos devem ser previstos para avaliar, controlar e fazer essas modificações.

Dessa forma, há a necessidade de direcionar esforços para a manutenção de software, pois muitos dos existentes são de 10 a 15 anos atrás. Ainda que eles tenham sido desenvolvidos utilizando as melhores práticas de projeto (*design*) e implementação na época, os critérios para "bom" software eram diferentes dos atuais, pois a preocupação era de tamanho do código fonte e espaço de armazenamento. Vários desses sistemas têm sido migrados para novas plataformas, adaptados às mudanças de equipamentos e de tecnologia e incrementados para atender às novas necessidades dos usuários. Nessas ocasiões, depara-se com estruturas e lógicas pobremente projetadas, código fonte nem sempre organizados e documentação deficiente dos sistemas de software que devem continuar funcionando [Pressman, 2009]. A manutenibilidade de software tende a ser negligenciada, em decorrência de alguns fatores, por exemplo: i) anseios das organizações responsáveis pelo seu desenvolvimento em atender rapidamente o mercado; ii) cultura de que a manutenção

não é uma atividade nobre; e iii) caracterização da atividade de manutenção como um problema a ser tratado posteriormente ao desenvolvimento dos produtos de software.

Manutenção de software é definida como o conjunto de atividades necessárias para o provimento de suporte a um sistema, minimizando esforços e custos. As atividades são executadas antes, na forma de planejamento, e após o software ser entregue ao cliente, na forma de correção de erros, adição de novas funções ou realização de adaptações a para novos ambientes e novas necessidades [Pigoski, 1996; Pressman, 2009; Calazans, 2005]. As atividades de manutenção de software são complexas e consomem em torno de 50% do custo total de um projeto, além de despender mais recursos que o próprio desenvolvimento [Coleman *et al.*, 1994]. Dessa forma, de fundamental importância para redução de custos e de recursos, a manutenção de software deve ser incluída no projeto e planejada desde o início do desenvolvimento, garantindo que o software desenvolvido seja manutenível. As atividades de manutenção de software podem ser organizadas em quatro categorias [Pressman, 2009]:

- **Manutenção Corretiva.** É realizada quando são corrigidos erros não identificados durante os testes;
- **Manutenção Adaptativa.** É realizada quando o software precisa ser adaptado às novas tecnologias (hardware e software) implantadas no ambiente operacional;
- **Manutenção Evolutiva.** É realizada quando o software deve englobar novos requisitos ou melhorias decorrentes da evolução na tecnologia de implementação empregada.
- **Manutenção Preventiva.** É realizada quando o software é alterado para aumentar sua manutenibilidade e/ou confiabilidade. Esse tipo de manutenção é raro em ambientes de desenvolvimento. Modificações realizadas neste tipo de manutenção não afetam o comportamento funcional do produto de software.

Uma das características de qualidade de um sistema de software é a Manutenibilidade. Manutenibilidade é a facilidade que sistemas têm de receber manutenção, pois são "mais" fáceis de serem compreendidos, corrigidos, adaptados, acrescidos de funções e melhorados. Além disso, para a realização da manutenção, há necessidade de menos recursos, tempo e pessoal. São muitas as causas que levam um software a ter baixa manutenibilidade e terem dificuldades na realização de manutenções, por exemplo, displicência com o projeto, escrita de código, realização de teste e

documentação, disponibilidade futura do pessoal desenvolvedor do software, uso de linguagens de programação obsoletas e planejamento das fases de manutenção.

A necessidade de cuidado e de atenção com projeto, codificação e testes é fundamental para o software ser manutenível; caso contrário, há aumento da probabilidade de ocorrência de erros e da instabilidade no sistema e provimento de documentação para o software ser entendido e manutenível. Por outro lado, é preciso que alguns participantes do desenvolvimento do software, no qual é realizada a manutenção, estejam disponíveis para ajudar no entendimento e na compreensão do software. A falta de planejamento da fase de manutenção é uma causa que impacta diretamente na manutenibilidade do software. O descaso com esse planejamento gera falhas não previstas e difíceis de corrigir.

3.3. Processo de Manutenção de Software

Os processos de manutenção de software são responsáveis pela definição do conjunto de tarefas fundamentais para o atendimento eficiente das solicitações de manutenção de software [Aparecido, 2011]. Por causa da preocupação crescente em torno da manutenção de software, foram criadas normas e técnicas para apoiarem e direcionarem os processos de manutenção de software. Essas normas buscam o aperfeiçoamento e a melhoria da qualidade de manutenibilidade de sistemas de software, bem como a padronização de técnicas e de métodos para realização de manutenção de software.

3.3.1. ISO/IEC 12207

A ISO/IEC 12207 é uma norma internacional, criada em 1995, que estabelece um cenário comum para os processos de desenvolvimento e de ciclo de vida de software. Seu objetivo é definir processos de ciclo de vida, tarefas e atividades para auxiliar a comunicação, o entendimento e a compreensão entre os compradores de produtos e de serviços de software e os desenvolvedores, os mantenedores, os fornecedores, entre outros envolvidos e interessados no ciclo de vida do produto de software [ISO/IEC, 2008]. Essa norma foi criada com o intuito de apoiar as organizações a compreenderem melhor os procedimentos pertinentes à aquisição e ao fornecimento de um produto de software e aperfeiçoar e agilizar a execução de projetos [Machado, 2006]. Essa norma é compatível com qualquer modelo de ciclo de vida, técnica de engenharia de software ou linguagem de programação, pois ela não é composta por tecnologias, ferramentas ou medidas. Dessa

forma, ela é empregada mundialmente e acompanha a evolução da engenharia de software [Gesing, 2011].

Na ISO/IEC 12207, são definidas atividades relacionadas ao ciclo de vida de software e classificadas em sete grupos de processos. Os processos de ciclo de vida são descritos por sua finalidade e resultados esperados, bem como as atividades e as tarefas necessárias para alcançar esses resultados. Essa divisão dos grupos de processos de ciclo de vida de software é ilustrada na Figura 3-1.

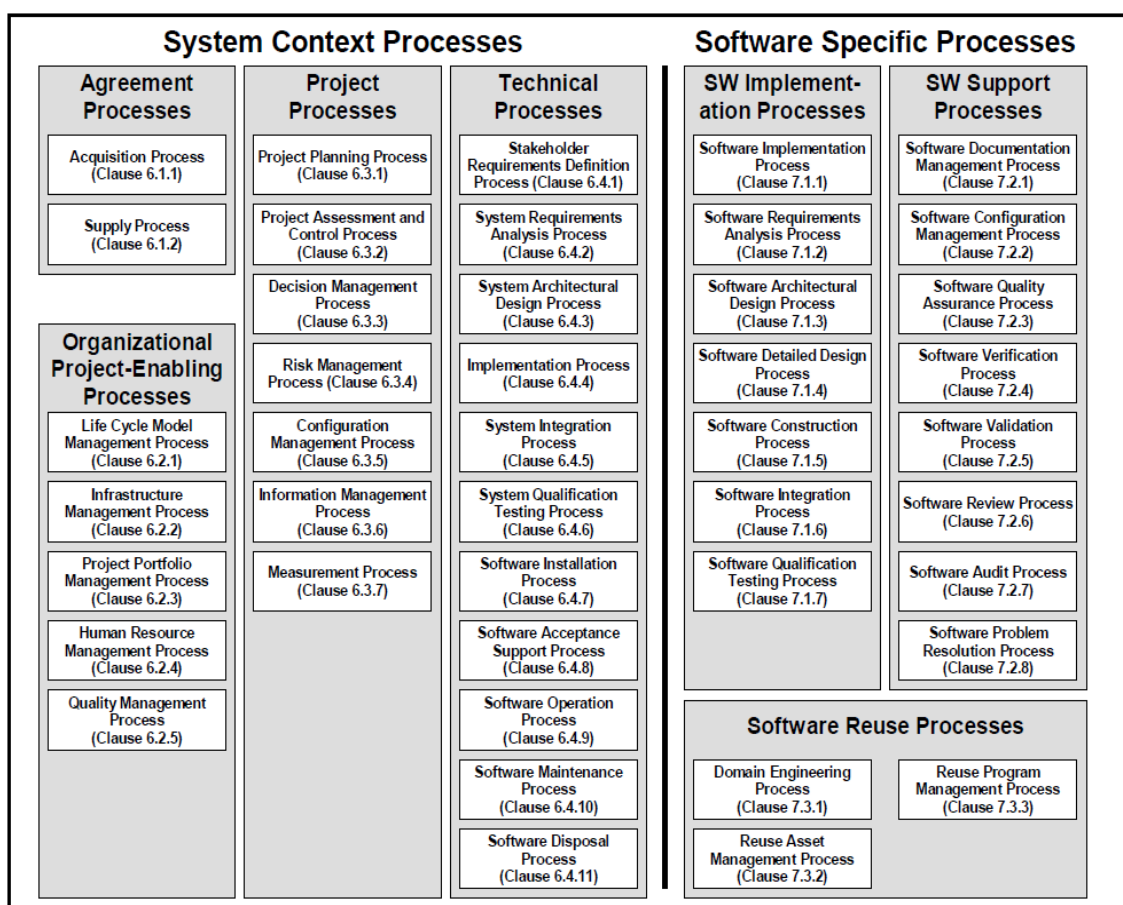


Figura 3-1 - Grupos de Processos de Ciclo de Vida de Software (Fonte: [ISO/IEC, 2008])

É importante descrever, de forma geral, a ideia do grupo Processos Técnicos, uma vez que os processos pertinentes à área manutenção de software fazem parte dos Processos Técnicos. De acordo com a Figura 3-1, o grupo Processos Técnicos contém processos que abordam desde a aquisição de requisitos de um software à sua descontinuação. Esses processos são responsáveis pela definição de atividades, com foco organizacional, que buscam reduzir riscos nas tomadas de decisões. Além disso, essas atividades permitem que os produtos possuam qualidade, por exemplo, produtividade, usabilidade, funcionalidade

e robustez, importante para a conclusão de uma aquisição. Eles atentam para que os produtos e os serviços estejam em conformidade com as exigências impostas pela sociedade, tais como, exigências ambientais, sanitárias e de segurança [ISO/IEC, 2008]. Como visto, o processo Manutenção de Software é relatado pela ISO/IEC 12207 como pertencente ao grupo Processos Técnicos e suas principais atividades são (Figura 3-2) [Paduelli, 2007]:

- **Implantação do processo.** Define tarefas necessárias para a organização de procedimentos referentes à manutenção de software. Inclui procedimentos de recebimento e de monitoração de pedidos de manutenção, estabelecendo uma interface com o processo de configuração. Dentro dessa atividade, verifica-se o escopo da manutenção, bem como a equipe a ser contratada para realizar a manutenção, observando suas obrigações e os recursos disponíveis;
- **Análise do problema e da modificação.** Essa atividade visa analisar a requisição de manutenção, com objetivo de classificar e de definir tamanho, custo e tempo necessários para sua realização. Outras atividades são relacionadas ao desenvolvimento e à documentação das modificações feitas;
- **Implantação da modificação.** Inclui os elementos que precisam ser modificados e como será feita a mudança. No final, é realizada validação para confirmar se as partes modificadas foram alteradas com sucesso e não houve alteração em itens não modificados;
- **Revisão/Aceitação da modificação.** Aborda mais especificamente a questão da validação. O cliente verifica se a manutenção foi realizada da forma desejada e se o software atende as suas necessidades, após as modificações realizadas;
- **Migração.** Acontece quando o software é implantado em novo ambiente, exigindo novas adaptações. Essa atividade aborda elementos do ambiente original e do ambiente novo para obter o resultado dos impactos após a mudança de ambiente;
- **Descontinuação do software.** O software é descontinuado a pedido do cliente.

Essa norma tem assumido papel importante nas instituições, pois as ajudam a definir padrões e processos de ciclo de vida de seus sistemas de software. Além de ser bem definida e facilitar os passos iniciais do desenvolvimento de sistemas de software, ela é uma referência utilizada pelo modelo de maturidade de software MPS [Softex, 2012].

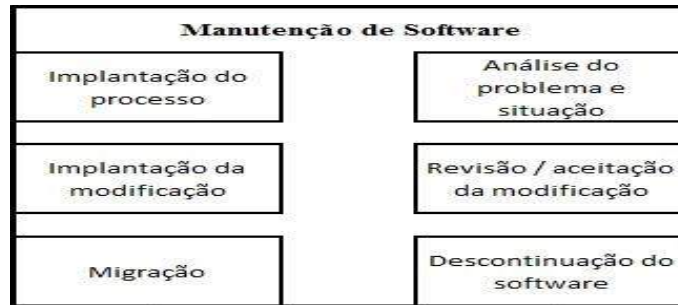


Figura 3-2 - Atividades de Manutenção de Software (Fonte: [Paduelli, 2007])

3.3.2. ISO/IEC 14764

Descrita como uma extensão da ISO/IEC 12207, a ISO/IEC 14764 propõe um processo detalhado de manutenção, cujo objetivo é apoiar o planejamento, a estruturação, o gerenciamento e a tomada de decisões relacionados às atividades de manutenção de software. Essa norma aborda critérios desde as etapas de planejamento e logística a descontinuação do software, em que acontece a retirada do software do ambiente de trabalho. Além disso, ela relata detalhadamente as atividades a serem realizadas nos processos de software, bem como seus objetivos e seus resultados esperados. Por fim, a ISO/IEC 14764 propõe sugestões e observações para orientar os envolvidos e garantir que os objetivos sejam alcançados [ISO/IEC, 2006].

As atividades pertinentes ao processo de manutenção sugerido pela ISO/IEC 14764 acontecem em duas etapas. A primeira etapa é relacionada às atividades iniciais de desenvolvimento e visam, por exemplo, à troca de informações e de conhecimentos entre o pessoal envolvido e o grupo responsável pela manutenção. Na segunda fase, o foco é garantir que o software seja funcional e atenda as necessidades do usuário [Aparecido, 2011]. O ciclo de atividades que integram o processo de manutenção de software é (Figura 3-3) [ISO/IEC, 2006]:

- **Processo de implementação.** A equipe responsável pela manutenção define os planos e os procedimentos que devem ser seguidos durante o processo de manutenção;
- **Análise de modificações e problemas.** Atividades que ocorrem após a realização de alterações no software. Elas são requisitadas quando problemas são encontrados;
- **Implementação da modificação.** Ao decorrer da atividade de modificação, a equipe responsável pela manutenção realiza alterações e testes no software;

- **Revisão da manutenção/aceitação.** Essa atividade verifica se os métodos utilizados são indicados, além de garantir que as alterações feitas estão corretas, de acordo com os padrões estabelecidos;
- **Migração.** A equipe de manutenção determina as ações necessárias para realizar a migração do sistema, além de definir e documentar as etapas necessárias;
- **Retirada.** O software, ao chegar ao fim de sua vida útil, deve ser "aposentado". Isso acontece por causa de alguns fatores, por exemplo, uso tecnologias ultrapassadas ou desenvolvimento de novos produtos que detenham novas tecnologias e sejam mais fáceis de realizar manutenção.

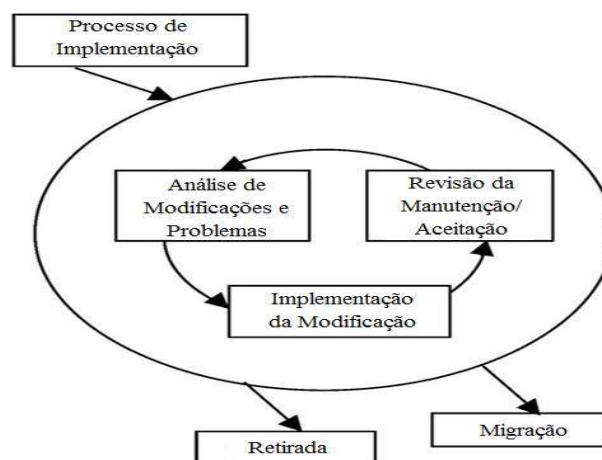


Figura 3-3 - Processos de Manutenção de Software (Fonte: [ISO/IEC, 2006])

3.3.3. Melhorias no Processo de Software

Entende-se como melhorias de processos de software o conjunto de esforços envolvidos na evolução e no aperfeiçoamento desses processos. Essa evolução ocorre por causa de estudos, de análises, de medições e de melhor utilização das práticas de Engenharia de Software aplicadas nos sistemas de software durante seu ciclo de vida [Pressman, 2009]. Dessa forma, as empresas buscam apoio em modelos de maturidade de processos, os quais são utilizados para medir e para classificar a eficiência dos processos pertencentes a uma empresa ou organização.

3.3.3.1. *Software Maintenance Maturity Model*

O *Software Maintenance Maturity Model* (S3M) é um modelo de maturidade das atividades pertinentes à manutenção de software [April; Abran, 2009]. Esse modelo utiliza práticas definidas por ISO/IEC 12207 [ISO/IEC, 2008] e ISO/IECO 14764 [ISO/IEC, 2006]. Os objetivos desse modelo podem ser resumidos em assegurar maior

disponibilidade e estabilidade do sistema, a confiança transmitida pela equipe de software perante os usuários e a agilidade na correção de falhas. Em conformidade com o S3M, as empresas têm seus processos avaliados e o nível de maturidade desses processos é enquadrado em um dos seis níveis de maturidade apresentados por esse modelo. Esses níveis são definidos como [April; Abran, 2009]:

- **Nível 0 - Incompleto.** Organizações nesse nível de maturidade apresentam modelo de processo incompleto ou podem não ter modelo;
- **Nível 1 - Executado.** Os processos são executados, porém não são documentados. Com isso, o sucesso da atividade depende das habilidades individuais dos profissionais envolvidos;
- **Nível 2 - Gerenciado.** A importância do uso do processo é observada por alguns membros da equipe. Algumas atividades definidas pelo processo são seguidas;
- **Nível 3 - Estabelecido.** A empresa possui os processos definidos e suas práticas são documentadas e entendidas pelos membros envolvidos;
- **Nível 4 - Previsível.** As práticas são documentadas e executadas formalmente de acordo com as metas definidas;
- **Nível 5 - Otimizado.** Além de ter as atividades executadas formalmente e corretamente documentadas, as empresas nesse nível de maturidade apresentam revisões regulares e propostas de melhoria do processo, visando à adaptação em caso de modificações.

Com isso, nesse modelo de maturidade, são apresentadas características necessárias para garantir o sucesso dos projetos relacionados à manutenção de software, bem como a satisfação da equipe desenvolvedora e do usuário.

3.3.3.2. *Corrective Maintenance Maturity Model*

Nesse modelo, é proposta uma estrutura de avaliação dos níveis de maturidade de uma empresa, baseada na sua capacidade em atender as solicitações de manutenção corretiva [Kajko-Mattsson, 2001]. O Modelo de Maturidade da Manutenção Corretiva (C3M) pode ser descrito em Manutenção Corretiva, cujas atividades são para corrigir erros e Maturidade do Processo de Manutenção Corretiva, cuja maturidade das empresas apresenta tarefas como organização, definição, gerenciamento, medição e controle de seu processo de manutenção de software. Essa maturidade impacta diretamente na agilidade

que a empresa apresenta na resolução de problemas de software e na aplicação do processo de manutenção corretiva [IEEE, 2004].

Os aspectos de maturidade propostos pelo CM3 e na visão de manutenção corretiva são referentes às atividades [Aparecido, 2011] Pré-entrega, Resolução de Problemas, Testes e Documentação das Correções e Treinamento das Equipes que atendem solicitações corretivas. Dessa forma, os processos das organizações são avaliados e elas são classificadas e enquadradas pelo CM3 em um de seus três níveis de maturidade [IEEE, 2004]:

- **Nível 1 - Inicial.** As organizações não detêm controle ou documentação sobre os processos de solicitações corretivas. Com isso, a realização das atividades fica dependente exclusivamente das habilidades individuais dos profissionais da empresa;
- **Nível 2 - Definido.** As empresas nesse nível de maturidade possuem modelos básicos de processo e, embora esse modelo apresente suas práticas com baixo nível de detalhamento, as empresas possuem controle e *feedback* sobre as solicitações de manutenção corretiva;
- **Nível 3 - Otimizado.** As empresas possuem total controle e *feedback* das solicitações corretivas. Dessa forma, elas possuem conhecimento sobre as atividades específicas do processo, além de estarem preocupadas com processos de melhorias.

3.4. Considerações Finais

Neste capítulo, foram abordados definições, conceitos e modelos da área de conhecimento de Manutenção de Software. Esses termos foram apresentados detalhadamente, tentando esclarecer qualquer dúvida ou má compreensão, tendo em vista a importância de entender esses conceitos teóricos e o jogo computacional educacional englobar esse conteúdo.

4. TRABALHOS RELACIONADOS

SPARSE (*Software Project semi-Automated Tool for Software Engineering*) [Souza *et al.*, 2010] é um jogo de simulação baseado em regras. O jogador é capaz de interagir com vários ambientes relacionados à Engenharia de Software. As atividades enfrentadas pelos jogadores são modelo de processo, projeto de software, gerente de projetos, desenvolvedores e ferramentas. As suas regras foram criadas baseadas no conjunto de boas práticas da Engenharia de Software. Durante o jogo, há eventos aleatórios, em que o jogador depara-se com situações novas e desafiadoras. O objetivo é completar o desenvolvimento do projeto, preservando prazos e custos definidos com o cliente, atentando para a qualidade do software.

No jogo SimSE [Navarro; Hoek, 2009], o objetivo é complementar a aprendizagem de alunos de Engenharia de Software. É um jogo educativo de simulação de processos de Engenharia de Software, em que o jogador tem a oportunidade de exercer o papel de gerente de projetos. A sua função é delegar atividades aos integrantes da equipe, monitorar processos, utilizar ferramentas disponíveis, contratar e demitir pessoal e seguir prazos. Ao decorrer do jogo, o jogador deve tomar decisões, as quais podem influenciar positiva ou negativamente no resultado final do projeto. No final, o jogador recebe pontuação e informações sobre o seu desempenho no jogo.

TestEG [Oliveira, 2013] é um jogo de simulação 3D que busca aprimorar o processo de ensino-aprendizagem de Teste de Software. No início, o jogador é um gerente de testes, recebe um orçamento e deve contratar sua equipe, composta por 3 funcionários. Ao decorrer do jogo, o gerente de testes pode solicitar a seus funcionários realizarem treinamentos para aumentar conhecimento e contribuições à empresa. O jogo possui o módulo Administrador, o usuário realiza manutenção de dados cadastrais e de perguntas, e o módulo Jogador, o usuário joga, consultando suas regras e o *ranking* de pontuação. O objetivo é responder 10 perguntas sem "estourar" orçamento e prazo previstos.

SimulES [Figueiredo *et al.*, 2007] é um jogo de tabuleiro destinado ao apoio do aprendizado das atividades e responsabilidades de um gerente de software. No SimulES, os jogadores disputam a liderança no desenvolvimento, na implementação e na entrega do projeto. Os componentes do jogo são tabuleiro, cartas, cartas de projetos e um dado. A

dinâmica ocorre com o lançamento de dados e o recebimento de cartas de acordo com o número obtido no dado, podendo retirar três cartas problemas e conceitos, caso o valor obtido no dado seja entre um e três; ou três cartas problemas e cartas conceitos e cartas engenheiro, caso seja entre quatro e seis. As cartas engenheiros possuem opções para construir artefatos, inspecionar artefatos, corrigir defeitos e integrar artefatos em módulos. O cliente realiza validação, executando verificação aleatória dos módulos em busca de possíveis erros. O jogador é vencedor caso a validação ocorra com sucesso e não sejam encontrados defeitos nos módulos de seu projeto.

AprendES [Feitosa; Campos, 2010] foi desenvolvido para auxiliar a aprendizagem de Engenharia de Software e teve como base o jogo educacional SimulES. O jogo é composto por tabuleiro e cartas e jogado por quatro jogadores. As cartas do jogo são do tipo problemas, conceitos, qualidade, tamanho, complexidade, *status* e nível de experiência. No início do jogo, são definidas especialidades de cada engenheiro de software, orçamento, complexidade e quantidade de artefatos. Após a definição inicial, cada jogador retira uma carta problema e uma carta conceito. O sucesso do jogo é alcançado quando o jogador consegue a solução do problema dado a ele no prazo e no orçamento estipulados.

Simultrain [Sts, 2013] aborda a simulação de gestão de projetos de software, com base no PMBoK (*Project Management Body of Knowledge*), expondo o jogador a situações em que é capaz de analisar, de compreender, de adaptar-se, de tomar decisões e de resolver problemas, complementando seu conhecimento básico e prático. O jogador tem como principais tarefas o controle de cronogramas e a definição de atividades ao pessoal da equipe de projeto. Ao decorrer do jogo, o jogador é atualizado com o resultado de suas ações e, caso suas decisões tenham sido tomadas de forma equivocada, ele recebe um aviso que o incentiva para empenhar-se a corrigir o erro antes de prosseguir no jogo. O jogo tem suporte a multijogadores, limite de quatro jogadores e o público alvo é profissionais experientes em gestão de projetos de software. No jogo, professores podem criar cenários e adicionar informações, experiências, problemas e situações.

SE-RPG [Molléri, 2006] é um jogo computacional voltado para o aprendizado do processo de desenvolvimento de software, jogado por apenas um jogador que inicia o jogo escolhendo o gerente e o projeto a ser desenvolvido. O jogo desenvolve-se com a apresentação do escopo do projeto, do investimento e do prazo. O jogador deve escolher a

linguagem de programação a ser utilizada e um modelo de processo de desenvolvimento de software. A partir desse ponto, a função do jogador é montar equipes, designar tarefas, realizar alterações na equipe, caso algum membro esteja com problemas ou dificuldades, e acompanhar a evolução do projeto para concluir a entrega dentro do prazo previsto. O jogo termina com a apresentação de um relatório, baseado nas ações e nas decisões tomadas pelo jogador no decorrer do jogo, contendo os pontos obtidos por ele.

O jogo SESAM [Drappa; Ludewig, 2000] proporciona aos jogadores o contato com eventos e simulações que buscam estimular a aprendizagem de processos de Engenharia de Software de forma dinâmica. O usuário assume o papel de gerente de projetos, cujo objetivo é completar um projeto, atento ao cronograma, ao orçamento e à qualidade final do produto. Ao decorrer do jogo, o usuário tem o controle da simulação que permite alterar tarefas, funções e atividades, utilizando comandos textuais. O SESAM é jogado por um jogador, sendo destinado a usuários com experiência profissional na área de gestão de projetos de software. O produto final é apresentado ao cliente que analisa e diz se está ou não satisfeito com o produto entregue. Além do *feedback* do cliente, o jogador tem auxílio de uma ferramenta de análise para poder refletir sobre suas ações e decisões.

Embora os jogos apresentados anteriormente sejam educacionais, apenas os jogos SimulEs [Figueiredo *et al.*, 2007] e AprendES [Feitosa; Campos, 2010] utilizam o tabuleiro como plataforma. Os demais jogos, inclusive o MaintES, desenvolvido neste trabalho, são jogos computacionais. Assim, eles visam facilitar a aprendizagem dos alunos em diversas áreas de conhecimento. Entretanto, o MaintEs difere-se dos demais jogos pois envolve conhecimentos sobre a área de Manutenção de Software. É importante elucidar que, apesar das várias buscas não foram encontrados jogos em Manutenção de Software. Assim, a elaboração deste jogo visou contribuir como alternativa de estudos para estudantes dessa área.

5. MaintES - MAINTENANCE EDUCATIONAL SOFTWARE

5.1. Considerações Iniciais

Neste capítulo, é apresentado o jogo computacional educacional MaintES (*Maintenance Educational Software*), cujo objetivo é aprimorar o processo de ensino-aprendizagem na área Manutenção de Software. Com isso, são descritos e detalhados as especificações, a modelagem e o funcionamento do jogo.

As especificações do MaintES são apresentadas na Seção 5.2. A modelagem, o diagrama de casos de uso e o diagrama de Entidade-Relacionamento do MaintES são descritos na Seção 5.3. O funcionamento do MaintES é mostrado na Seção 5.4.

5.2. Especificações do Software

No MaintES, o jogador assume o papel de um estudante que busca vencer torneios de perguntas, em três diferentes níveis, cujos temas estão relacionados com Manutenção de Software. O estudante inicia o jogo no nível *Iniciante* sem pontos e pequena quantidade de dinheiro. Com isso, ele pode testar seu conhecimento, respondendo a perguntas referentes aos temas do seu nível atual para ganhar dinheiro e aumentar sua pontuação. Dessa forma, quando determinada quantidade de pontos for alcançada e o jogador possuir dinheiro suficiente, ele pode participar de um torneio pagando sua inscrição.

O dinheiro acumulado pode ser utilizado na contratação de funcionários para sua equipe. A equipe é importante, pois, com ela, o estudante pode eliminar alternativas falsas das perguntas dos torneios, facilitando a sua conquista. Assim, a evolução de nível acontece quando o jogador conquistar o torneio de seu nível atual e apresentar pontuação requerida para o nível seguinte. O sucesso no jogo ocorre quando o jogador vence os três torneios e acumula pontuação suficiente para alcançar o nível *Master*.

5.3. Modelagem

5.3.1. Diagrama de Casos de Uso

A funcionalidade do jogo MaintES é mostrada na Figura 5-1 utilizando o Diagrama de Casos de Uso da UML (*Unified Modeling Language*). Os seguintes casos de uso podem ser realizados pelo ator *Jogador*: i) Fazer *login*; ii) Consultar as regras do jogo; iii) Atualizar dados; iv) Ver *rank* das 5 maiores pontuações; v) Contratar membros para a

Equipe; vi) Demitir membros da Equipe; vii) Salvar o jogo; viii) Estudar; ix) Responder perguntas dos Treinamentos; x) Responder pergunta dos Torneios; xi) Utilizar sua Equipe nos Torneios; xii) Conferir suas conquistas; e xiii) Fazer *logout*. O ator Administrador pode realizar os seguintes casos de uso: i) Fazer *login*; ii) Gerenciar Jogadores; iii) Gerenciar Administradores; iv) Gerenciar perguntas dos Treinamentos; v) Gerenciar perguntas dos Torneios; vi) Alterar as configurações do Jogo; e vii) Fazer *logout*.

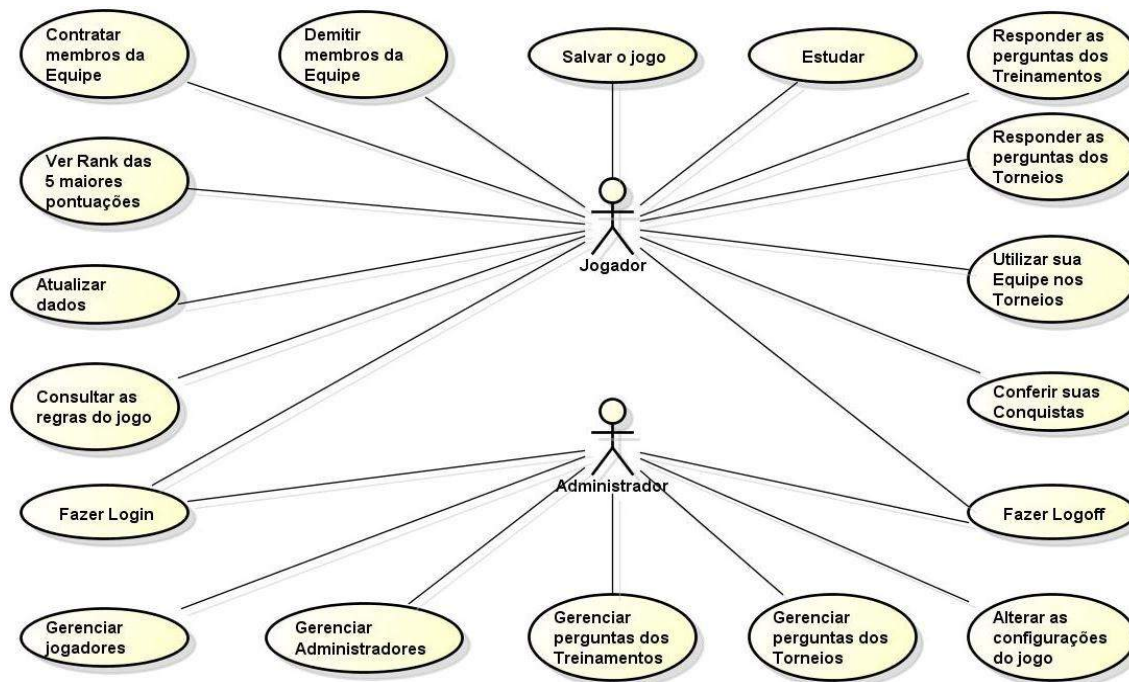


Figura 5-1 - Diagrama de Casos de Uso

5.3.2. Modelo Relacional

Foi utilizado o Modelo Relacional para representar as tabelas na modelagem dos dados utilizados no MaintES (Figura 5-2). As informações contidas nessas tabelas são fundamentais para o funcionamento do jogo, uma vez que elas possuem informações dos usuários, dos administradores, das perguntas dos três níveis de dificuldade de treinamento e de torneio, gerais e do *rank*.

5.4. Funcionamento

As características do jogo computacional educacional MaintES e seu funcionamento são detalhados nesta seção.

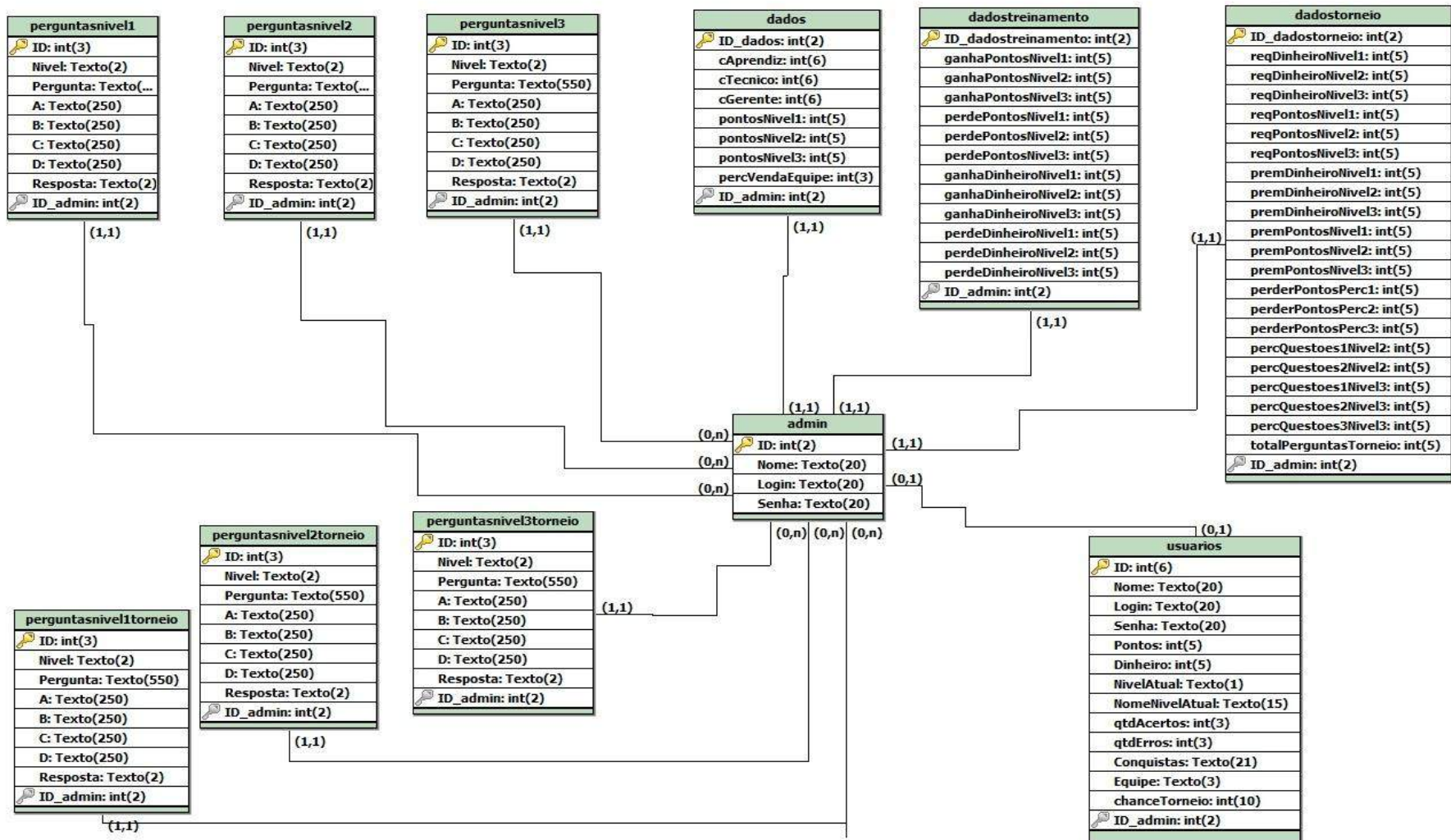


Figura 5-2 - Modelo Relacional

5.4.1. Organização Modular

O MaintES é um jogo computacional educacional, para um jogador, em que ele desempenha o papel de um estudante de Engenharia de Software que busca melhorar seu nível de conhecimento sobre Manutenção de Software. Com isso, durante o jogo, o seu conhecimento é testado na participação de torneios de perguntas. Para ter sucesso nesses torneios, o jogador deve responder perguntas sem errar. O MaintES possui dois módulos:

- **Módulo Administrador.** Esse módulo permite aos Administradores alterarem dados cadastrais de jogadores e de administradores e realizarem manutenção nas perguntas e nas configurações gerais do MaintES;
- **Módulo Jogador.** Esse módulo permite aos jogadores realizarem alteração em seus dados cadastrais, acessarem o *rank* e a ajuda e jogarem, estudando, respondendo perguntas e contratando membros para sua equipe.

5.4.2. Módulo Administrador

Para acessar o MaintES como Administrador, deve-se realizar autenticação no MaintES fornecendo *login* e senha (Figura 5-3). Em seguida, deve-se selecionar a opção **Administrador** e a opção **Entrar**. Uma vez realizada a autenticação, a tela principal de Administrador é mostrada (Figura 5-4), contendo as ações possíveis que os administradores podem executar: i) Gerenciar Jogadores; ii) Gerenciar Administradores; iii) Gerenciar Perguntas; iv) Configurações; e v) Sair.

Na opção **Gerenciar Jogadores**, o Administrador pode incluir, alterar e remover Jogador (Figura 5-5). As contas de Jogador podem ser encontradas utilizando o sistema de busca pelo *login* ou navegando nos índices de Jogador, utilizando os controles de navegação. O Administrador precisa estar atento ao salvar os dados cadastrais de Jogador, pois os campos não podem estar vazios. Além disso, o campo **Senha** e o campo **Confirmar Senha** precisam ser iguais e a informação do campo **Login** não pode ser a mesma de um *login* existente. Com isso, uma mensagem de confirmação de sucesso em verde é apresentada; caso contrário, uma mensagem de erro em vermelho é apresentada.

Caso o Administrador opte por **Gerenciar Administradores**, uma tela é apresentada (Figura 5-6), contendo opções para incluir, alterar e remover

Administrador do MaintES. Assim como no **Gerenciar Jogadores**, as contas de Administrador podem ser encontradas utilizando o sistema de busca pelo *login* ou navegando nos índices de Administrador, utilizando os controles de navegação.



Figura 5-3 - Tela de Entrada do Jogo - Autenticação



Figura 5-4 - Tela Inicial do Administrador



Figura 5-5 - Tela de Alteração de Dados Cadastrais de Jogadores



Figura 5-6 - Tela de Alteração de Dados Cadastrais de Administrador

O Administrador precisa estar atento ao realizar alterações nos dados cadastrais de Administrador, pois os campos não podem estar vazios. Além disso, o campo **Senha** e o campo **Confirmar Senha** precisam ser iguais e a informação do campo **Login**

não pode ser a mesma de um *login* existente. Com isso, uma mensagem de confirmação de sucesso em verde é apresentada; caso contrário, uma mensagem de erro em vermelho é apresentada.

Na opção **Gerenciar Perguntas**, o Administrador pode incluir, alterar e excluir perguntas dos três níveis do treinamento e do torneio (Figura 5-7). O Administrador pode navegar nas perguntas selecionando o modo (Treinamento ou Torneio), nível (Iniciante, Amador ou Profissional) utilizando as setas de navegação. Para o correto armazenamento das perguntas, o Administrador precisa preencher o campo **Pergunta**, os campos referentes a cada alternativa e selecionar a alternativa correta. É importante lembrar que, ao incluir ou modificar uma pergunta, os campos não podem estar vazios. Caso isso ocorra, uma mensagem de erro em vermelho é apresentada; caso contrário, uma mensagem em verde é apresentada confirmando o sucesso da operação.

ATUALIZAR PERGUNTAS

=====
ID Pergunta: 1 IR

Modo:

Nível:

Pergunta: Dentre os 4 diferentes tipos de manutenção de software, qual é aquela responsável pela busca e resolução de erros, falhas e defeitos encontrados em um produto de software já entregue ao cliente?

a)

b)

c)

d)

Alternativa correta: A) B) C) D)

Figura 5-7 - Tela de Gerenciamento de Perguntas

Se o Administrador optar por **Configurações**, é apresentada uma tela contendo informações sobre **Configurações Gerais**, **Configurações do Treinamento** e **Configurações do Torneio**. Caso o Administrador selecione a aba **Configurações Gerais** (Figura 5-8), ele pode modificar as faixas de níveis, alterando os pontos necessários para cada nível (valores limites das faixas não podem se sobrepor). Ainda, ele pode

modificar o custo de contratação de membros da equipe e alterar a porcentagem do valor recebido pelo Jogador, caso demita algum membro de sua equipe. Depois de realizar as alterações, o Administrador pode **Salvar**; caso contrário, ele pode recuperar os valores originais ao optar por **Padrão**.



Figura 5-8 - Tela de Configurações Gerais

Na aba **Configurações do Treinamento** (Figura 5-9), o Administrador é capaz de realizar modificações nos valores ganhos e perdidos de dinheiro e de pontos dos três diferentes níveis de treinamento. Da mesma forma, como nas **Configurações Gerais**, após as alterações, o Administrador pode **Salvar**; caso contrário, ele pode recuperar os valores originais ao utilizar a opção **Padrão**.

Na aba **Configurações do Torneio** (Figura 5-10), o Administrador pode alterar os pontos requeridos, o valor da inscrição e o total de perguntas (máximo de perguntas permitidas é o menor entre o total de perguntas dos três torneios) dos três níveis. Ainda, ele pode modificar a porcentagem de pontos que o Jogador perde caso erre alguma pergunta e falhe no torneio. O Administrador pode definir a probabilidade de sorteio de uma pergunta do nível Iniciante ou Amador no Torneio Amador (máximo permitido para o nível Iniciante é 30%) e a probabilidade de sorteio de uma pergunta do nível Iniciante, Amador ou Profissional no Torneio Profissional (máximo

permitido para o nível Iniciante e Amador são 10% e 30%, respectivamente). Assim, se as alterações estiverem de acordo com o interesse do Administrador, ele pode **Salvar**; caso contrário, ele pode recuperar os valores originais ao utilizar a opção **Padrão**.



Figura 5-9 - Tela de Configurações dos Treinamentos



Figura 5-10 - Tela de Configurações dos Torneios

É importante lembrar que os campos de porcentagem não aceitam caracteres, recebendo apenas valores entre 0 e 100 e os outros campos não aceitam caracteres ou números negativos.

5.4.3. Módulo Jogador

Para jogar o MaintES, o Jogador precisa realizar sua autenticação, fornecendo *login* e senha previamente cadastrados pelo Administrador. O Jogador deve selecionar a opção **Jogador** e a opção **Entrar**. Uma vez realizada a autenticação, a tela principal de Jogador é mostrada (Figura 5-11) e ele pode escolher entre as opções: i) Jogar; ii) Regras; iii) Atualizar dados; iv) *Rank*; e v) Sair. Caso o Jogador opte por **Jogar**, ele é direcionado para o cenário do jogo, um ambiente doméstico, onde ocorrem as ações do jogo (Figura 5-12).



Figura 5-11 - Tela Inicial do Jogador

Para consultar as regras, o Jogador deve escolher a opção **Regras** (Figura 5-13). Para realizar modificações em seus dados cadastrais como *login*, senha e nome, o Jogador deve escolher a opção **Atualizar dados**; ele deve estar atento para o novo *login* não ser o mesmo de um existente e os campos não estejam vazios (Figura 5-14). Uma mensagem de confirmação aparece caso as mudanças sejam efetuadas com sucesso; caso contrário, uma mensagem de erro é apresentada. O Jogador pode escolher a opção **Rank**

para consultar o *rank* das cinco maiores pontuações do MaintES (Figura 5-15) ou a opção **Sair** para sair do jogo.



Figura 5-12 - Ambiente do MaintES

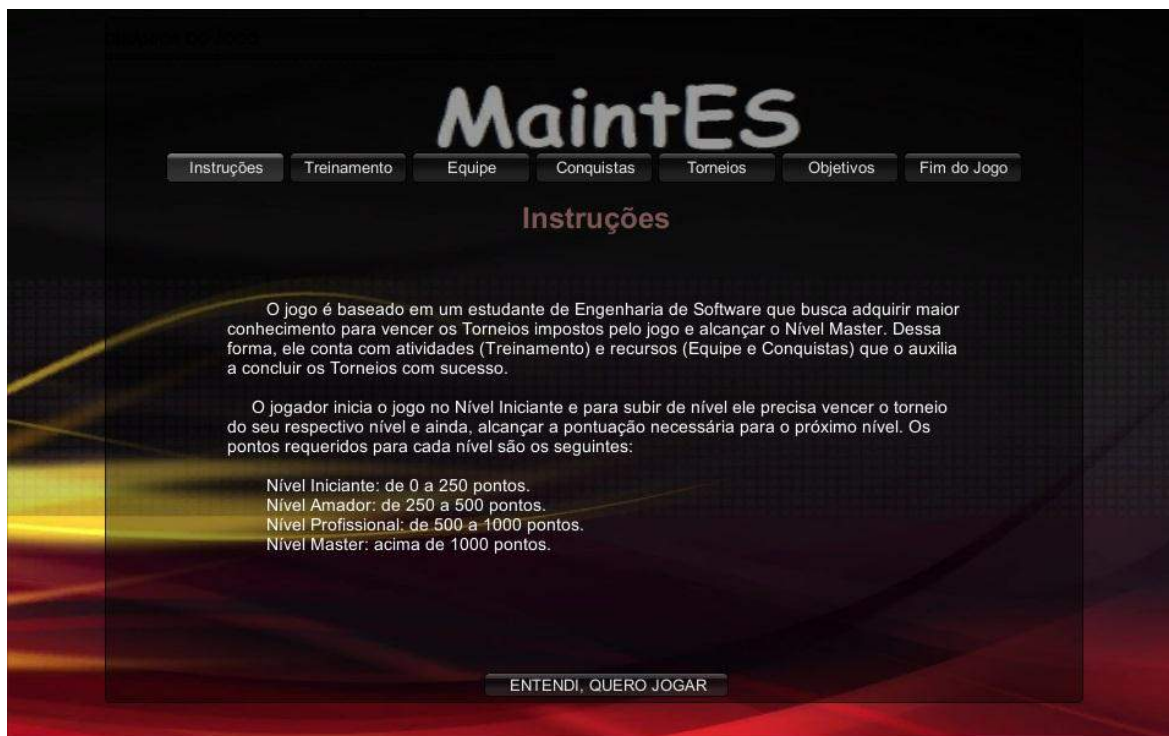


Figura 5-13 - Regras do MaintES



Figura 5-14 - Tela para Atualizar Dados Cadastrais do Jogador



Figura 5-15 - Rank

Ao iniciar o jogo, o Jogador assume o papel de um estudante em um ambiente doméstico em que ele pode escolher as opções: i) Estudar; ii) Participar do Treinamento; iii) Participar do Torneio; iv) Verificar suas Conquistas; v) Salvar; vi) Gerenciar sua

Equipe; e vii) Consultar a ajuda do MaintES. Ao optar por **Estudar** (ir até a **Prateleira de Livros**), o Jogador é direcionado para a tela apresentada na Figura 5-16, na qual tem acesso a texts resumos de cada nível.

Os textos para o nível **Iniciante** são Tipos de Manutenção de Software, Definições de Manutenção de Software e Software Legado. Os textos para o nível **Amador** são Manutenibilidade, Processo de Manutenção de Software e Norma ISO/IEC 12207. Os textos para o nível **Profissional** são Engenharia Reversa, Norma ISO/IEC 14764 e CMMM. Caso queira ler algum resumo, o Jogador deve escolher a opção **Estudar**; com isso, é apresentada uma tela contendo o texto resumo sobre o assunto escolhido (Figura 5-17). Nessa tela, o Jogador pode estudar sobre o assunto e navegar diretamente para os outros textos resumos. O Jogador pode (re)ler um resumo, sempre que julgar necessário. É importante lembrar que, quanto mais preparado ele estiver para enfrentar o Treinamento e os desafios dos Torneios, ele perde menos pontos e, conseqüentemente, melhor é seu desempenho no decorrer do jogo.

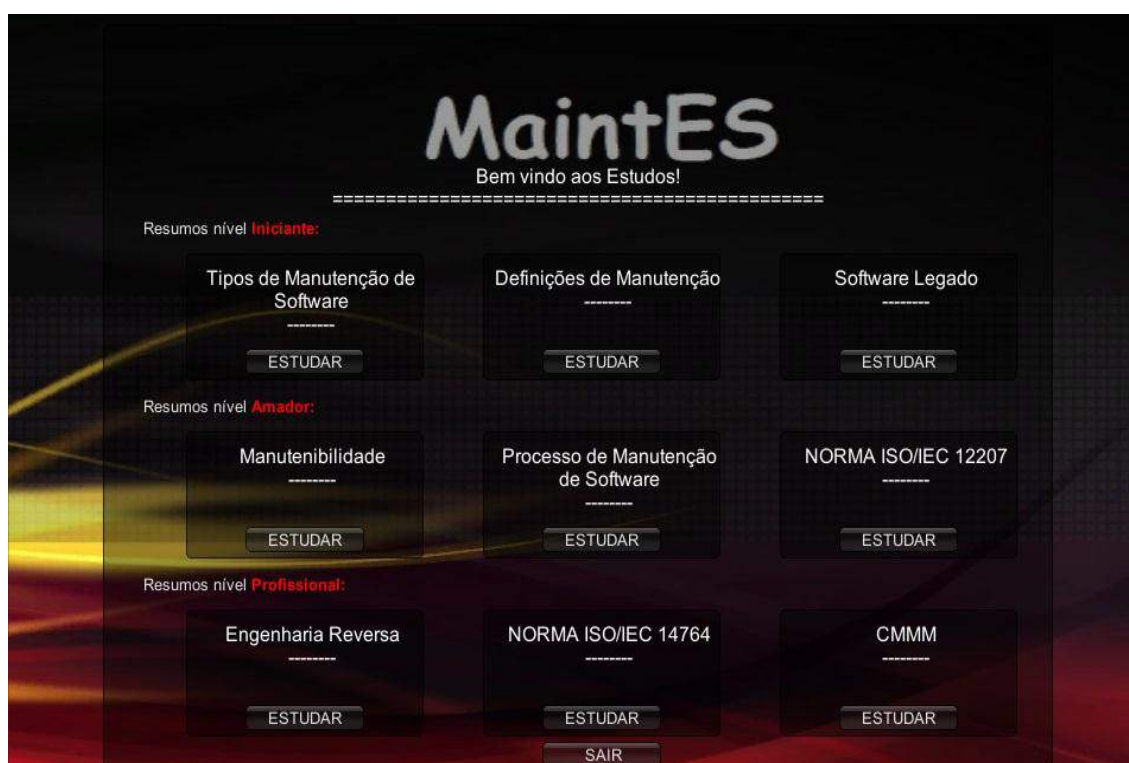


Figura 5-16 - Tela de Estudo

No **Treinamento** (Figura 5-18), o Jogador é desafiado a responder perguntas pertencentes ao seu nível atual. Ele responde perguntas, selecionado a alternativa que julgar correta e, caso acerte, recebe pontos e dinheiro, previamente configurados, para o

referente nível (os valores recebidos são customizáveis pelo Administrador que podem ser consultados na opção **Ajuda**, na aba **Treinamento**). Caso não responda corretamente, o Jogador perde pontos e dinheiro (previamente definidos pelo Administrador) referentes ao nível atual. Os valores ganhos e perdidos são: i) Nível Iniciante: 5 pontos e 10 reais; ii) Nível Amador: 10 pontos e 20 reais; e iii) Nível Profissional: 50 pontos e 100 reais.



Figura 5-17 - Textos Resumos

Para o Jogador acessar o treinamento, ele deve ir até a **Mesa de Estudos**. Quando ele quiser sair, ele deve selecionar a opção que julgar correta para a pergunta atual e escolher a opção **Responder e Sair**, pois, se o Jogador selecionar a opção **Sair**, é considerado que ele errou a pergunta e perde pontos e dinheiro referentes ao seu nível atual. O Jogador pode, sempre que desejar, consultar as conquistas que possui (Figura 5-19). As conquistas são metas pré-determinadas que servem de incentivo. Caso o Jogador alcance alguma dessas metas, ele recebe recompensas na forma de pontos e de dinheiro referente à meta conquistada (Tabela 5-1).

Uma meta pode ser alcançada uma única vez, com exceção dos Torneios. O acesso às conquistas é feito quando o Jogador vai até a **Prateleira de Troféus**. Um dos recursos que o Jogador dispõe para auxiliá-lo durante os torneios é a **Equipe**, sendo

composta por três tipos de membros: i) Aprendiz; ii) Técnico; e iii) Gerente. A equipe pode possuir vários membros de cada tipo, sendo que cada membro pode ser "utilizado" apenas uma vez. Cada membro é capaz de eliminar alternativas falsas das perguntas do **Torneio**: i) Aprendiz: Elimina uma alternativa falsa; ii) Técnico: Elimina duas alternativas falsas; e iii) Gerente: Elimina três alternativas falsas.

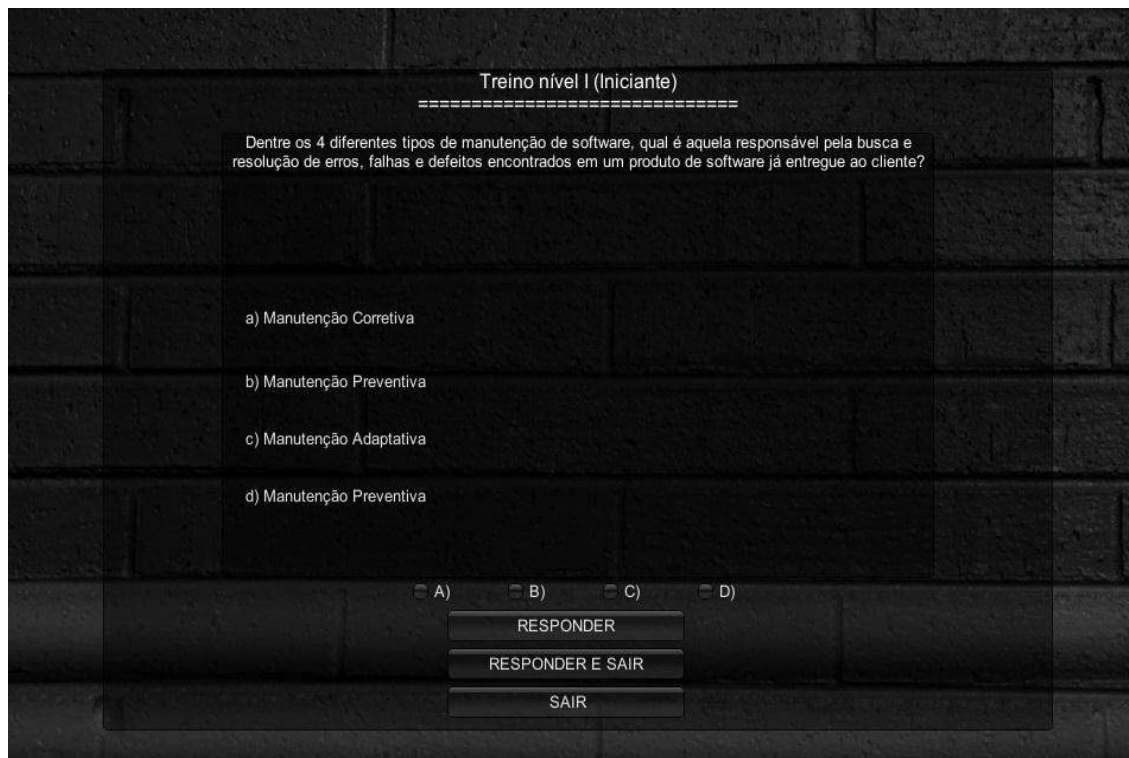


Figura 5-18 - Treinamento

Caso o Jogador escolha a opção **Equipe**, no painel principal do Jogador (Figura 5-12), ele é direcionado para a tela apresentada na Figura 5-20. Nessa tela, o Jogador gerencia sua equipe, contratando e demitindo membros. Os valores para contratação são: i) Aprendiz: 250 reais; ii) Técnico: 500 reais; e iii) Gerente: 750 reais. No caso do Jogador escolher por demitir algum membro, uma porcentagem do valor gasto na contratação é reembolsada. Essa porcentagem é, por padrão, 50%, podendo ser alterada pelo Administrador.

O objetivo do jogo é alcançar o nível **Master**. Para isso, o Jogador precisa alcançar 1.000 pontos, que pode ser alterado pelo Administrador, e vencer os **Torneios** em seus três níveis de dificuldade. Para participar de um **Torneio**, o Jogador deve ir até a **Mesa de Computador** para conhecer os requisitos para a sua entrada nos **Torneios** (Figura 5-21). Caso possua os requisitos, o Jogador pode participar do

Torneio ao escolher a opção **Iniciar Torneio** (Figura 5-22). Os requisitos são: i) Torneio nível Iniciante: 100 pontos e valor da inscrição: 100 reais; ii) Torneio nível Amador: 200 pontos e valor da inscrição: 200 reais; e iii) Torneio nível Profissional: 500 pontos e valor da inscrição: 500 reais.



Figura 5-19 - Tela de Conquistas

Tabela 5-1 - Conquistas e Recompensas

Reais	Recompensas	Pontos	Recompensas	Acertos	Recompensas
200 reais	5 pontos e 10 reais	50 pontos	5 pontos e 10 reais	5 acertos	5 pontos e 10 reais
400 reais	8 pontos e 15 reais	100 pontos	10 pontos e 20 reais	10 acertos	10 pontos e 20 reais
600 reais	15 pontos e 25 reais	250 pontos	20 pontos e 40 reais	20 acertos	20 pontos e 40 reais
800 reais	25 pontos e 40 reais	500 pontos	50 pontos e 80 reais	50 acertos	40 pontos e 80 reais
1.000 reais	25 pontos e 40 reais	750 pontos	80 pontos e 160 reais		
1.500 reais	80 pontos e 150 reais	1.000 pontos	160 pontos e 320 reais		
2.500 reais	150 pontos e 200 reais	1.500 pontos	320 pontos e 640 reais		

Na tela de entrada no **Torneio** (Figura 5-21), é apresentada ao Jogador a chance dele concluir com sucesso o **Torneio**. Essa chance é calculada com base na quantidade de perguntas que o Jogador acerta durante o **Treinamento**. À medida que acerta mais perguntas durante o **Treinamento**, suas chances de vencer o **Torneio** aumentam. Assim, o Jogador pode usar esse recurso para traçar sua estratégia de forma eficiente; por exemplo, não entrar no **Torneio** caso sua chance de sucesso seja baixa. Cabe ressaltar que, quando "sobe" ou "desce" de nível, o Jogador tem suas chances "zeradas".

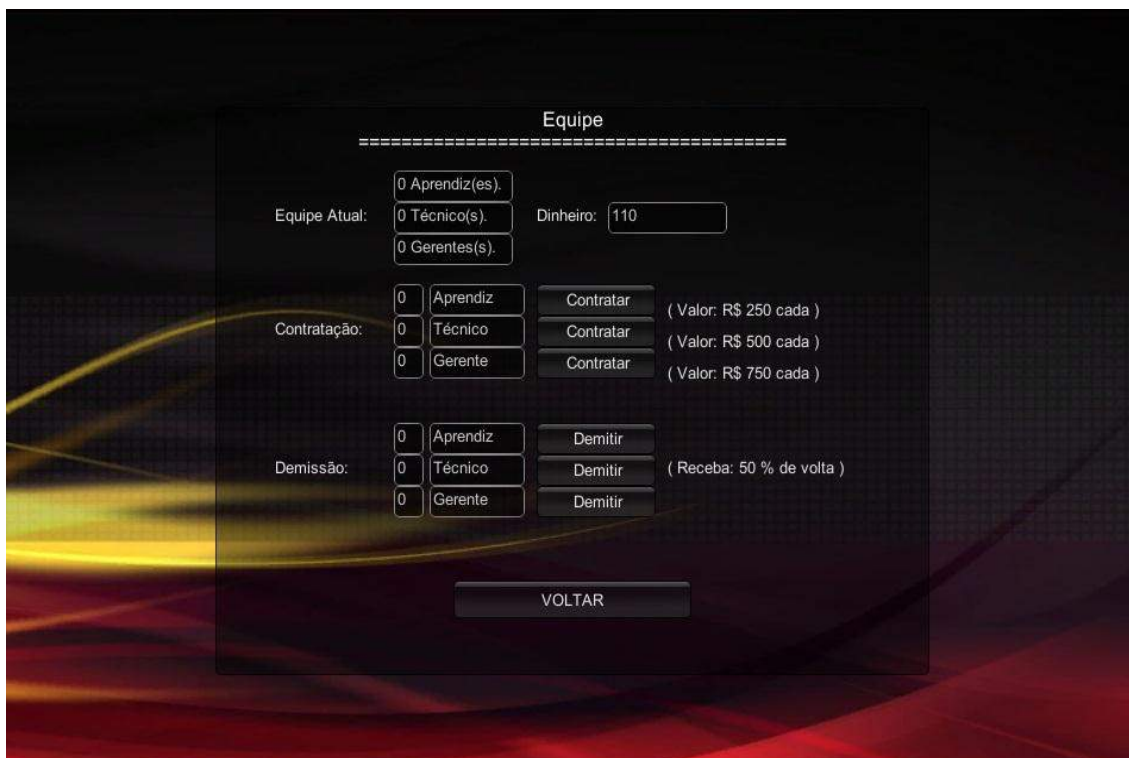


Figura 5-20 - Tela de Gerenciamento da Equipe



Figura 5-21 - Entrada no Torneio

Para vencer um torneio, o Jogador precisa responder, sem erro, todas as perguntas do **Torneio** correspondente ao seu nível. Para isso, ele conta com a ajuda dos membros de sua **Equipe** que o auxiliam eliminando alternativas falsas das perguntas do Torneio (Figura

5-22). Dessa forma, ao vencer um **Torneio**, o Jogador recebe o prêmio e a conquista correspondente ao nível disputado. Os prêmios são: i) Nível Iniciante: 80 pontos, 200 reais e troféu; ii) Nível Amador: 200 pontos, 500 reais e troféu; e iii) Nível Profissional: 400 pontos, 1000 reais e troféu. Caso o Jogador erre alguma pergunta durante o **Torneio**, ele é eliminado e perde uma porcentagem de seus pontos: i) 20% - Torneio nível Iniciante; ii) 30% - Torneio nível Amador; e iii) 40% - Torneio nível Profissional.



Figura 5-22 - Torneio Iniciante e Janela da Equipe

Para auxiliar o Jogador, o jogo possui uma tela de ajuda (Figura 5-13) contendo regras e informações sobre os componentes do jogo (**Instruções, Treinamento, Equipe, Conquistas, Torneios, Objetivos e Fim do Jogo**). O Jogador pode consultá-la quando tiver dúvidas ou para obter informações sobre o jogo. O acesso a essa ajuda pode ser: i) no painel principal do Jogador; ii) no *menu* Principal do jogador; iii) na entrada do Treinamento; e iv) na entrada do Torneio. O fim do jogo acontece de duas formas: i) o Jogador vence o jogo (ao alcançar o Nível Master) (Figura 5-23); e ii) o Jogador perde o jogo (caso fique sem pontos e sem dinheiro) (Figura 5-24). Mas, em ambos os casos, é permitido ao Jogador reiniciar o jogo com os valores iniciais.



Figura 5-23 - Tela Apresentada ao Jogador que Venceu o Jogo



Figura 5-24 - Tela Apresentada ao Jogador que Perdeu o Jogo

5.5. Considerações Finais

O jogo computacional educacional MaintES mostra-se uma ferramenta capaz de aprimorar o processo de ensino-aprendizagem no tema Manutenção de Software, uma vez

que trabalha habilidades específicas do tema e estratégia, planejamento e competição. Ainda, com a utilização desse jogo, instigante e desafiador, em ambientes escolares, os estudantes estão sujeitos a aprender de modo interativo e divertido, evoluindo seu conhecimento.

No decorrer do MaintES, o jogador é testado, a todo momento, com perguntas sobre o assunto de Manutenção de Software. O administrador possui o controle das perguntas e pode, a qualquer momento, reformulá-las para tornar o conteúdo do jogo mais objetivo, direcionando as perguntas para suas áreas de interesse. Cabe ao professor estudar e elaborar as perguntas de modo que se tornem mais eficaz para o ambiente escolar.

6. AVALIAÇÃO

6.1. Considerações Iniciais

A avaliação do MaintES foi feita por meio de um questionário, em que foram verificados a ludicidade, a motivação e o conteúdo sobre o tema Manutenção de Software. Foram convidados 78 avaliadores que, após terem jogado e concretizado seus julgamentos a respeito do jogo, responderam o questionário. Assim, foi possível coletar informações relevantes da utilização do MaintES no processo de ensino-aprendizagem em Manutenção de Software.

Discussão sobre os resultados obtidos é apresentada na Seção 6.2.

6.2. Resultados

A avaliação do MaintES contou com a colaboração de algumas pessoas que atuaram como avaliadores, o que possibilitou obter opiniões de diferentes pontos de vistas e de diferentes níveis de experiência. Os avaliadores foram estudantes de graduação do curso de Ciência da Computação de uma Instituição de Ensino Superior e professores e pesquisadores de Engenharia de Software.

O questionário foi elaborado e disponibilizado na ferramenta SurveyMonkey¹. Seu principal objetivo foi coletar informações específicas por meio de itens sobre Ludicidade, Sentimento do Usuário e Manutenção de Software. Os avaliadores poderiam avaliar esses itens como: i) concordo fortemente; ii) concordo; iii) discordo; e iv) discordo fortemente. Com esse questionário, foram obtidas informações de caracterização do avaliador, por exemplo, idade, sexo, escolaridade, conhecimento em manutenção de software e software educacional e experiência em manutenção de software e software educacional.

Dos 78 avaliadores convidados, 34 jogaram o MaintES e o avaliaram utilizando o questionário disponibilizado. A primeira parte do questionário foi destinada para a caracterização dos avaliadores: i) faixa etária (Figura 6-1); ii) sexo (Figura 6-2); e iii) escolaridade (Figura 6-3). O resultado dessa primeira parte foi 6 avaliadores (17,65%) têm idade entre 18 e 22 anos, 18 avaliadores (52,94%) têm idade entre 23 e 30 anos e 10 avaliadores (29,41%) têm idade acima de 30 anos. Além disso, 28 participantes

¹ <http://pt.surveymonkey.com/>

(82,35%) são do sexo masculino e 6 participantes (17,65%) são do sexo feminino. Quanto ao grau de escolaridade, 8 avaliadores (23,53%) possuem doutorado, 2 avaliadores (5,88) possuem mestrado, 7 avaliadores (20,59) possuem graduação e 17 avaliadores (50,00%) cursam graduação. Com essa etapa de caracterização, foi possível observar que a maioria dos avaliadores possui entre 23 e 30 anos, é do sexo masculino e são graduandos.

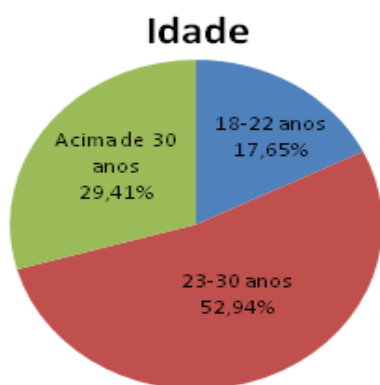


Figura 6-1 - Idade dos Avaliadores

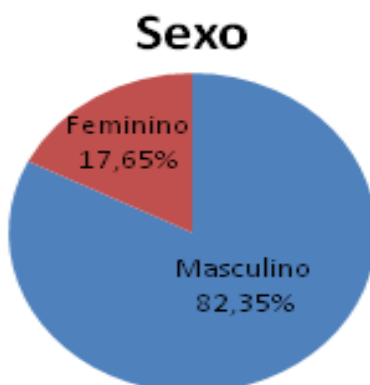


Figura 6-2 - Sexo dos Avaliadores

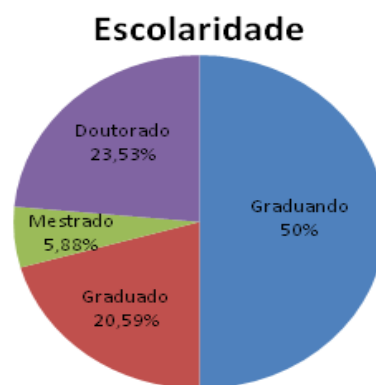


Figura 6-3 - Escolaridade dos Avaliadores

Para avaliar o nível de experiência dos avaliadores, foram elaboradas duas perguntas: i) conhecimento do usuário em Manutenção de Software; e ii) conhecimento do usuário em Software Educacional. Os participantes responderam essas questões escolhendo uma das seguintes opções: i) Iniciante; ii) Moderado; e iii) Avançado. Sobre o conhecimento em Manutenção de Software (Figura 6-4), 15 avaliadores (44,12%) alegaram ter conhecimento no nível Iniciante, 11 avaliadores (32,35%) alegaram ter conhecimento no nível Moderado e 8 avaliadores (23,53%) alegaram ter conhecimento no nível Avançado. Em relação ao conhecimento sobre Software Educacional (Figura 6-4), 13 avaliadores (38,24%) alegaram ter conhecimento no nível Iniciante, 17 avaliadores (50,00%) alegaram ter conhecimento no nível Moderado e 4 avaliadores (11,76%) alegaram ter conhecimento no nível Avançado. Com os resultados obtidos, pode-se perceber que a maioria dos avaliadores apresenta conhecimento Iniciante sobre Manutenção de Software e conhecimento Moderado sobre Software Educacional.

Além disso, a experiência dos avaliadores em Manutenção de Software e em jogar/desenvolver um Software Educacional foi avaliada (Figura 6-5). Com relação à

experiência sobre Manutenção de Software, os avaliadores responderam escolhendo uma das seguintes opções: i) teórica (aula, livro, etc...); e ii) prática (trabalho, atividades realizadas, etc...). Sobre a experiência em Software Educacional, os avaliadores responderam escolhendo uma das seguintes opções: i) utilização (jogar); e ii) desenvolvimento. Analisando o resultado, pode-se perceber que 30 avaliadores (88,24%) possuem experiência teórica em Manutenção de Software e 16 avaliadores (47,06%) têm experiência prática em Manutenção de Software. Além disso, 31 avaliadores (91,18%) jogaram um Software Educacional e 8 avaliadores (23,53%) participaram no desenvolvimento de um Software Educacional.

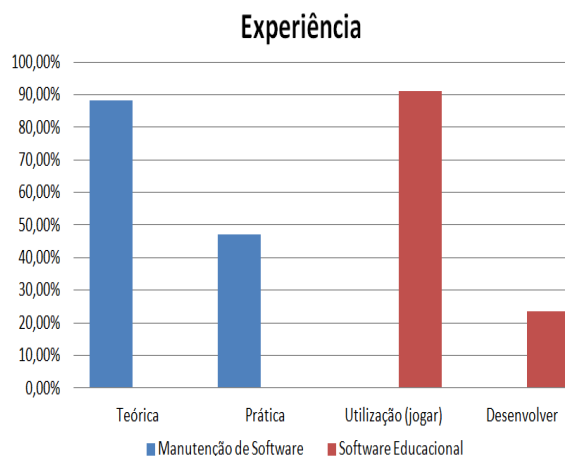
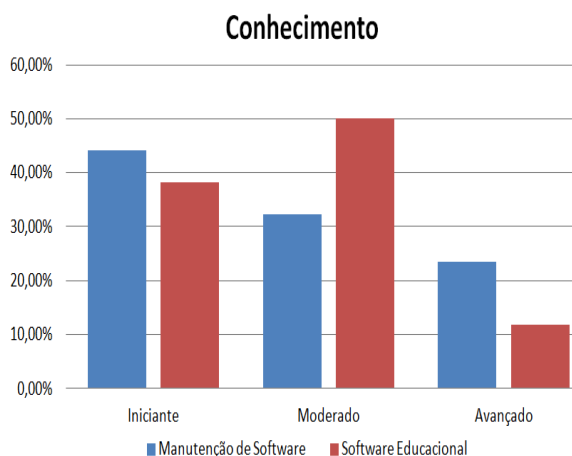


Figura 6-4 - Conhecimento dos Avaliadores

Figura 6-5 - Experiência dos Avaliadores

Conforme dito anteriormente, MaintES foi avaliado quanto à ludicidade, ao sentimento do usuário e ao conteúdo sobre manutenção de software. Para isso, foram elaboradas questões nas que os avaliadores responderam escolhendo uma das seguintes opções: i) Concordo Fortemente; ii) Concordo; iii) Discordo; e iv) Discordo Fortemente. Analisando o resultado apresentado na Figura 6-6, MaintES pode ser caracterizado como lúdico, pois a maioria dos avaliadores respondeu que "concordam" ou "concordam plenamente" que:

- MaintES manteve a motivação para continuar jogando e aplicando os conceitos (30 avaliadores - 88,23%);
- Ao terminar, os avaliadores sentiram-se realizado, satisfeito e com a certeza de acréscimo de conhecimento (25 avaliadores - 73,53%);

- MaintES possui interface atrativa e mantém a atenção dos jogadores (29 avaliadores - 85,29%).

Porém, 7 avaliadores (20,59%) "discordam" ou "discordam fortemente" que a interface do MaintES é atrativa e mantém a atenção dos jogadores, o que comprova que o MaintES pode ser melhorado.

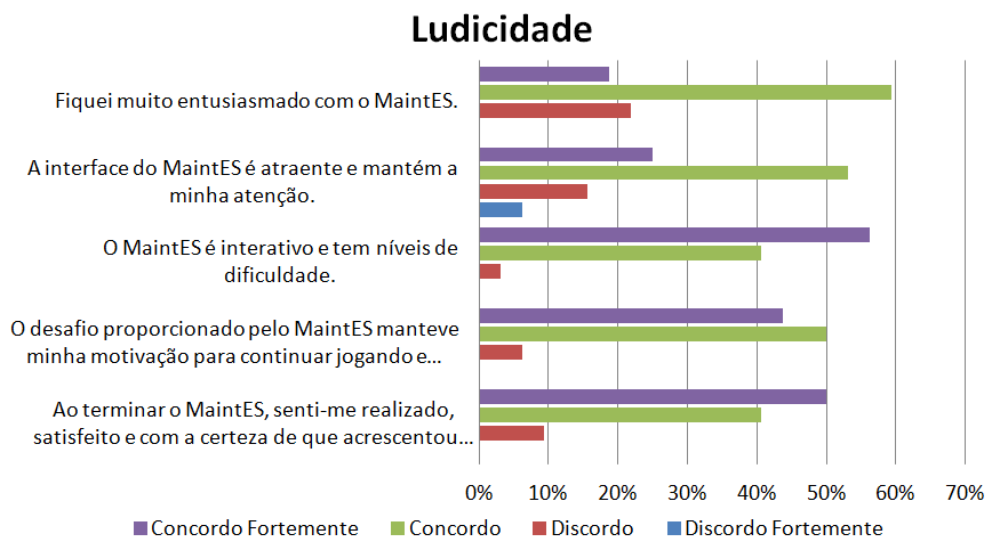


Figura 6-6 - Ludicidade do MaintES

Em relação ao sentimento do usuário (Figura 6-7), 29 avaliadores (85,29%) "concordam" ou "concordam fortemente" que o aprendizado foi bom após utilizar o MaintES e que proporciona concentração nas ações sobre Manutenção de Software, permitindo auxiliar no aprendizado. Isso é fundamental, pois uma importante característica dos jogos educacionais é a de proporcionar a concentração ao jogador, fazendo com que o aprendizado seja maior.

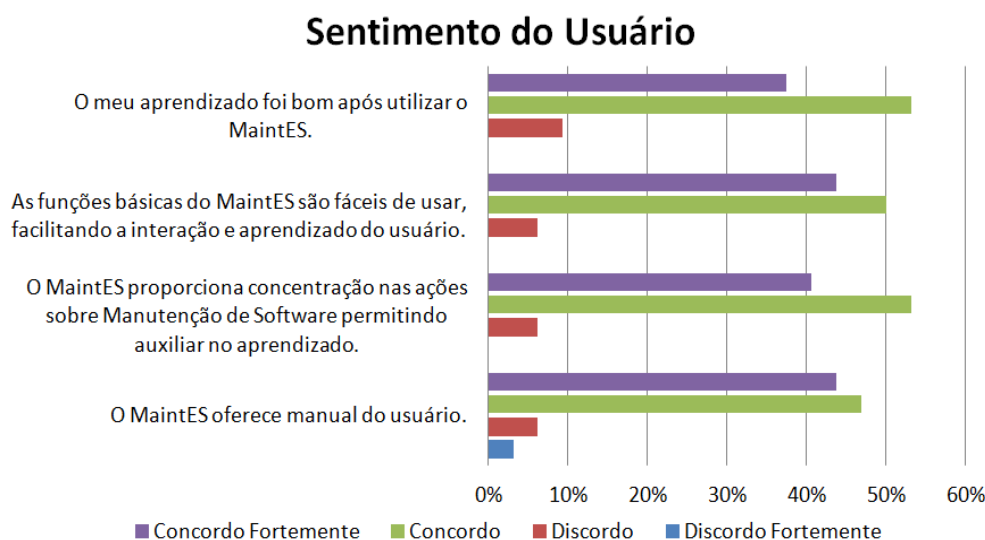


Figura 6-7 - Sentimento do Usuário em Relação ao MaintES

Sobre o assunto Manutenção de Software (Figura 6-8), 28 avaliadores (82,35%) responderam que o conteúdo de Manutenção de Software está bem organizado e fácil de entender no MaintES. Desses, 16 avaliadores (57,14%) "concordam fortemente" e 12 avaliadores (42,86%) "concordam". Ainda, 27 avaliadores (79,41%) "concordam" ou "concordam fortemente" que o resultado final do MaintES é satisfatório e reflete as ações realizadas durante o jogo.

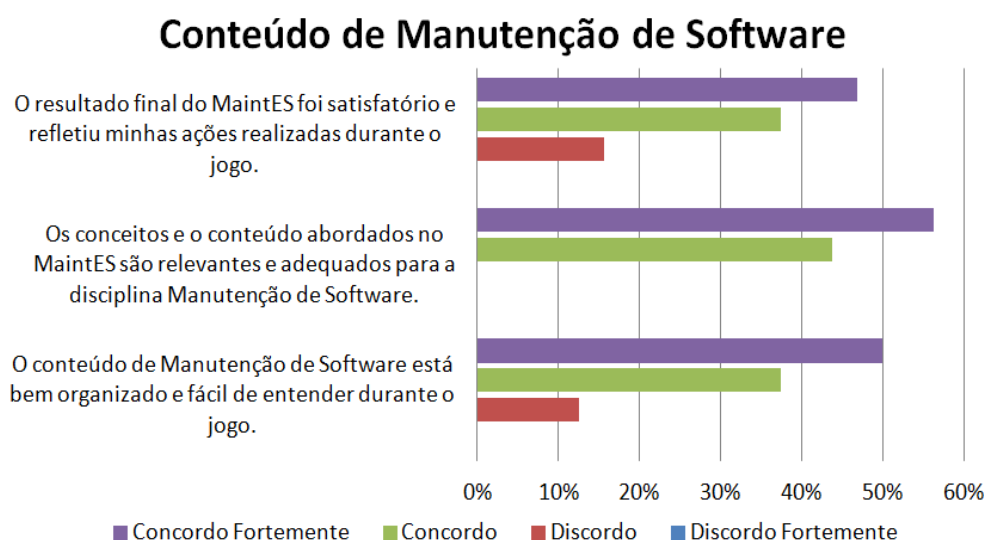


Figura 6-8 - Avaliação sobre o Conteúdo de Manutenção de Software

Por fim, foi realizada a análise da avaliação dos avaliadores com experiência avançada em Software Educacional (4 avaliadores). Todos responderam que o MaintES proporciona concentração nas ações sobre Manutenção de Software, permitindo auxiliar no aprendizado. Além disso, 3 avaliadores (75%) responderam que a interface do MaintES é atraente e mantém atenção do jogador e o conteúdo de Manutenção de Software está bem organizado e fácil de entender durante o jogo.

6.3. Considerações Finais

Analisando os resultados obtidos, pode-se concluir que o software educacional MaintES pode ser útil nos ambientes escolares e caracteriza-se como uma alternativa aos estudantes de Manutenção de Software que buscam uma opção de aprendizagem. Assim, a utilização do MaintES no ambiente escolar pode apoiar o processo de ensino-aprendizagem, pois "leva" o estudante a um ambiente interativo e desafiador, em que sua habilidade e seus conhecimentos em Manutenção de Software são testados.

A avaliação do MaintES possibilitou obter informações importantes a respeito das características da sua utilização. Os avaliadores puderam utilizar um "campo" no

questionário para emitir críticas e sugestões, colaborando para a avaliação e para a melhoria do MaintES. Esse *feedback* é importante, pois foi possível distinguir pontos de vistas distintos (estudante, pesquisador e professor). Assim, qualquer dificuldade ou melhoria proposta precisa ser considerada para ajustar o MaintES e disponibilizar nova versão. Essas críticas e sugestões foram analisadas e as mais relevantes, resumidamente, são:

- Utilizar mais artefatos gráficos e animações no "desenrolar" do jogo e na forma de ensinar como jogar, de modo a atrair mais os estudantes;
- Melhorar a interface dos textos de estudo e das perguntas nos treinamentos e nos torneios;
- Reestudar o cenário ambiente do jogo a fim de torná-lo mais profissional;
- Ter sempre visível a pontuação;
- Criar um guia, contendo os passos que o jogador deve seguir para completar o jogo.

7. CONSIDERAÇÕES FINAIS

7.1. Conclusões

Neste trabalho, foi possível analisar e estudar as principais dificuldades encontradas no processo de ensino-aprendizagem e possibilitou a percepção da importância em proporcionar aos estudantes uma alternativa de aprendizagem em que eles possam interagir. Dessa forma, eles são os principais responsáveis pelo desenrolar do jogo e o resultado é consequência exclusiva das ações realizadas pelo jogador durante o jogo e reflete o aprendizado adquirido por ele.

Considerando os resultados obtidos na avaliação do jogo educacional, realizado por meio de questionário específico, pode-se concluir que o MaintES obteve aceitação por partes dos avaliadores e sua utilização no ambiente escolar pode ser um facilitador no processo de ensino-aprendizagem. Desse modo, os profissionais responsáveis pela transmissão do conhecimento devem estudar a utilização de jogos educacionais, de modo que sua utilização possa reforçar o processo de ensino-aprendizagem e, conseqüentemente, incluí-los em seus métodos de ensino.

7.2. Contribuições

Este trabalho deixa como contribuição um software educacional - MaintES - cujo principal objetivo é apoiar o processo de ensino-aprendizagem sobre Manutenção de Software, inserindo os estudantes em um ambiente estimulador, com desafios, em que são exigidos conhecimentos em assuntos específicos sobre Manutenção de Software. Tendo em vista a carência de jogos educacionais voltados a essa área específica da Engenharia de Software e a dificuldade identificada no processo de ensino-aprendizagem, é esperado que os estudantes possam aprimorar seus conhecimentos em Manutenção de Software de maneira interativa e prazerosa com a utilização desse jogo.

7.3. Limitações

Algumas limitações verificadas no desenvolvimento do MaintES são:

- Impossibilidade de utilização do jogo, na versão Web, em plataformas Linux;
- Dificuldade de otimização do banco de dados com diversos usuários jogando simultaneamente;

- Dificuldade na implementação de um módulo adicional para competição *multiplayer*.

7.4. Trabalhos Futuros

Sugestões de trabalhos futuros, visando à continuidade deste trabalho, são:

- Implementação de um módulo competitivo *multiplayer*;
- Aprimoramento do modo de sorteio de perguntas de modo a diminuir a repetição de perguntas;
- Inclusão de novos temas referentes ao assunto Manutenção de Software;
- Aumento da base de dados de perguntas para diminuir a repetição de perguntas;
- Criação de mecanismos visuais para tornar o jogo mais fácil de ser compreendido;
- Permissão aos jogadores escolherem se querem participar do torneio do nível corrente ou de um de nível inferior ao corrente;
- Adição do painel de pontuação nos treinamentos e nos torneios.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aparecido, G. J. Avaliação e Formalização de um Processo para Agrupamento de Solicitações de Manutenção de Software. 99p. 2011.
- April, A.; Abran, A. A Software Maintenance Maturity Model (S3M): Measurement Practices at Maturity Levels 3 and 4. In: Electronic Notes Theoretical Computer Science. pp. 73-87. 2009.
- Borges, C. J. O Lúdico nas Interfaces das Relações Educativas. In: Revista de Pedagogia, v. 6. 2005.
- Calazans, A. T. S.; Oliveira, M. A. L. Avaliação de Estimativa de Tamanho para Projetos de Manutenção de Software. 2005.
- Catapan, A. H.; Plínio, C. F.; Souza, A. C.; Thomé, Z. R. C.; Cybis, W. D. A. Ergonomia em Software Educacional: A Possível Integração entre Usabilidade e Aprendizagem. In: Workshop sobre Fatores Humanos em Sistemas Computacionais. 1999.
- Coleman, D.; Lowther, B.; Oman, P. Using Metrics to Evaluate Software System Maintainability. v.27. pp. 44-49. 1994.
- Drappa, A.; Ludewig J. Simulation in Software Engineering Training. In: International Conference on Software Engineering (ICSE). pp. 199-208. 2000.
- Enricone, D. (Org.) Ser Professor. V. 6. 102p. 2008.
- Feitosa, A. C.; Campos, G. M. M. AprendES: Um Jogo Educacional para Auxiliar o Processo de Ensino-Aprendizagem da Engenharia de Software. In: Anais do Simpósio Brasileiro de Informática na Educação. 2010.
- Figueiredo, E.; Lobato, C. A.; Dias, K. L.; Leite, J. C.; Lucena, C. J. P. Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução. In: Sociedade Brasileira de Computação (SBC). pp. 37-46. 2007.
- Gesing, J. C. Melhoria do Processo de Software no Brasil: Utilização do Modelo Mps.Br em Empresas de Pequeno Porte. 92p. 2011.
- Gomes, A. S.; Castro Filho, J. A. de ; Gitirana, V.; Spinillo, A.; Alves, M.; Melo, M.; Ximenes, J. Avaliação de Software Educativo para o Ensino de Matemática. In: Workshop de Informática na Educação. 2002. v. 5. Disponível em: <<http://www.cin.ufpe.br/~asg/home.php?p=publications>>. Acessado em: 25/07/2011.
- Grando, R. C. O Jogo na Educação: Aspectos Didático-Methodológicos do Jogo na Educação Matemática. 2001. Disponível em: <www.cempem.fae.unicamp.br/lapemmec/cursos/el654/2001/jessica_e_paula/JOGO.doc>. Acessado em: 13/5/2012.
- IEEE. Guide to the Software Engineering Body of Knowledge (SWEBOK). 2004.

- ISO/IEC 12207. International Standard ISO/IEC 12207 Software Life Cycle Processes. 2008.
- ISO/IEC 14764. International Standard ISO/IEC/IEEE 14764 Software Engineering - Software Life Cycle Processes - Maintenance. 2006.
- Kajko-Mattsson, M. Towards a Business Maintenance Model. In: International Conference on Software Maintenance. pp 500-509. 2001.
- Krüger, S. E.; Fritsh, E. F.; Viccari, R. M. Avaliação Pedagógica do Software STR. In: Revista Brasileira de Informática na Educação. v. 8. pp. 21-33. 2001.
- Lopes, M. V. O.; Silva, V. M.; Araujo, T. L. Desenvolvimento Lógico-Matemático do Software ND. In: Revista Latino Americana de Enfermagem. v. 12. n. 1. pp. 92-100. 2004.
- Machado, C. F. Definindo Processos do Ciclo de Vida de Software usando a Norma NBR ISO/IEC 12207 e suas Ementas 1 e 2. 2006.
- Molléri, J. Utilizando o RPG como Ferramenta de Aprendizado para o Processo de Desenvolvimento de Software. 2006.
- Moratori, P. B. Por que Utilizar Jogos Educativos no Processo de Ensino Aprendizagem? 31p. 2003.
- Navarro, E.; Hoek, A.; Multi-Site Evaluation of SimSE. In: Technical Symposium on Computer Science Education. pp. 326-330. 2009.
- Oliveira, B. C. TestEG - Um Software Educacional para o Ensino de Teste de Software. 54p. 2013.
- Paduelli, M. M. Manutenção de Software: Problemas Típicos e Diretrizes para uma Disciplina Específica. 144p. 2007.
- Pigoski, T. M. Practical Software Maintenance: Best Practices for Managing Your Software Investment. 1996.
- Prensky, M. Digital Game-Based Learning. 442p. 2001.
- Pressman, Roger S. Software Engineering: A Practitioner's Approach. 860p. 2009.
- Prikladnicki, R.; Albuquerque, A. B.; Wangenheim, C. G.; Cabral, R. Ensino de Engenharia de Software: Desafios, Estratégias de Ensino e Lições Aprendidas. In: Fórum de Educação em Engenharia de Software. 2009.
- Schneider, J.; Johnston L. Extreme Programming at Universities - An Educational Perspective. In: International Conference on Software Engineering. 2003.
- Silva, G. M. O Uso do Computador na Educação, Aliada a Softwares Educativos no Auxílio ao Ensino e Aprendizagem. 2008. Disponível em:

<<http://www.educacaopublica.rj.gov.br/biblioteca/tecnologia/0021.html>>. Acessado em: 19/11/2013.

Smole, K. C. S.; Diniz, M. I. Ler, Escrever e Resolver Problemas: Habilidades Básicas para Aprender Matemática. Artmed. 204p. 2001.

Softex. Associação para Promoção da Excelência do Software Brasileiro - SOFTEX. MPS.BR - Guia Geral MPS de Software. Disponível em: <<http://www.softex.br>>. Acessado em: 22 de novembro de 2012.

Sommerville, I. Software Engineering. Addison Wesley. 792p. 2010.

Souza, M. M.; Resende, R. F.; Prado, L. S.; Fonseca, E. F.; Carvalho, F. A.; Rodrigues, A. D. SPARSE: Um Ambiente de Ensino e Aprendizagem de Engenharia de Software Baseado em Jogos e Simulação. In: Simpósio Brasileiro de Informática na Educação. 2010.

Sts. Website do Jogo. Disponível em: <www.sts.ch/smt7/demo/demo_e.htm>. Acessado em: 21/11/2013.

Tarouco, L. M. R.; Roland, L. C.; Fabre, M. J. M.; Konrath, M. L. P. Jogos Educacionais. v.2. 2004.

Valente, A. B. A Intransigência da Transferência de Conhecimento. FDE. 1993.

Vasconcellos, C. S. Construção do Conhecimento em Sala de Aula. 2004.

Wangenheim, C. G. von; Kochanski, D.; Savi, R. Revisão Sistemática sobre Avaliação de Jogos Voltados para Aprendizagem de Engenharia de Software no Brasil. In: Fórum de Educação em Engenharia de Software. pp. 41-48. 2009.