

UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



CARLOS DANILO CÂMARA DE OLIVEIRA

ENSINO DE GESTÃO DE RISCOS EM PROJETOS DE SOFTWARE USANDO JOGOS SÉRIOS, LÓGICA FUZZY E AGENTES INTELIGENTES

MOSSORÓ 2014

CARLOS DANILO CÂMARA DE OLIVEIRA

ENSINO DE GESTÃO DE RISCOS EM PROJETOS DE SOFTWARE USANDO JOGOS SÉRIOS, LÓGICA FUZZY E AGENTES INTELIGENTES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Marcos Evandro Cintra – UFERSA.
Co-Orientador: Prof. Dr. Francisco Milton Mendes Neto – UFERSA.

MOSSORÓ 2014

O conteúdo desta obra é de inteira responsabilidade de seus autores

Dados Internacionais de Catalogação na Publicação (CIP) Biblioteca Central Orlando Teixeira (BCOT) Setor de Informação e Referência

O48e Oliveira, Carlos Danilo Câmara de.

Ensino de gestão de riscos em projetos de *software* usando jogos sérios, lógica *fuzzy* e agentes inteligentes./ Carlos Danilo Câmara de Oliveira -- Mossoró, 2014. 116f.: il.

Orientador: Prof. Dr. Marcos Evandro Cintra

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal Rural do Semi-Árido. Pró-Reitoria de Pós-Graduação.

1. Gestão de riscos. 2. Jogos sérios. 3. Inteligência artificial. I. Título.

RN/UFERSA/BCOT /401-14 004.7 CDD:

Bibliotecária: Vanessa Christiane Alves de Souza Borba CRB-15/452

CARLOS DANILO CÂMARA DE OLIVEIRA

ENSINO DE GESTÃO DE RISCOS EM PROJETOS DE SOFTWARE USANDO JOGOS SÉRIOS, LÓGICA FUZZY E **AGENTES INTELIGENTES**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência Computação para a obtenção do título de Mestre em Ciência da Computação.

APROVADA EM: 14/02/2014.

Cintra - UFERSA

Presidente

Prof. Dr. Marcelino Pereira dos Santos Silva – UERN

Primeiro Membro

Prof. Dr. Romme Wtadimir de Lima - UERN

Segundo Membro

Profa. Dra. Ana Trindade Winck - UFSM

Membro Externo

RESUMO

A crescente demanda por profissionais de engenharia de software, especialmente gerentes de projeto, e a popularização da plataforma Web como catalisadora das relações humanas têm tornado esta plataforma interessante para ser utilizada no treinamento destes profissionais. Tem crescido também o uso de jogos como um instrumento atrativo no processo de ensino e aprendizagem. No entanto, o projeto de um jogo Web instrutivo que preencha os principais requisitos pedagógicos e técnicos para a formação de um gerente de projeto não é uma tarefa trivial. Nesse sentido, este trabalho foca no projeto de um jogo persistente baseado em navegador voltado para a gestão de riscos como componente de apoio no processo de qualificação de profissionais de gestão de projetos, chamado SuperRisk. O SuperRisk usa conceitos de Aprendizagem Baseado em Projetos. Agentes Inteligentes e Sistemas Fuzzy Baseados em Regras para permitir ao jogador adquirir experiência em um contexto real de gestão de projetos. O SuperRisk foi testado e validado por alunos de informática e computação. O SuperRisk tem sido bem recebido por professores e alunos, por simular o ambiente de gestão de riscos de forma coerente com a realidade e atraente para os alunos.

LISTA DE FIGURAS

Figura 1 – Cone de aprendizagem de Dale	. 11			
Figura 2 – Função de pertinência triangular	. 35			
Figura 3 – Função de pertinência trapezoidal	. 35			
Figura 4 – Função de pertinência S	. 36			
Figura 5 – Função de pertinência Gaussiana	. 36			
Figura 6 – Função de pertinência unitária	. 36			
Figura 7 – Exemplo de partições fuzzy com diferentes granularidades	. 39			
Figura 8 – Variável lingüística Temperatura				
Figura 9 – Raciocínio fuzzy clássico	. 49			
Figura 10 – Raciocínio fuzzy geral	. 50			
Figura 11 – Diagrama de atividades	. 59			
Figura 12A – Gráficos de Fuzzyficação e Defuzzyficação	. 68			
Figura 13B – Gráficos de Fuzzyficação e Defuzzyficação	. 69			
Figura 14 – Tela de Login do SUPERRISK	. 85			
Figura 15 – Tela de edição de cadastro	. 86			
Figura 16 – Tela de acompanhamento do projeto	. 87			
Figura 17 – Tela de classificação dos jogadores	. 88			
Figura 18 - Tela com o resumo do Mercado de Trabalho disponível no projeto par	а			
todos os jogadores	. 89			
Figura 19 – Tela que mostra a Equipe do jogador	. 90			
Figura 20 – Tela de Aquisições do projeto	. 91			
Figura 21 - Tela do Módulo de registro dos eventos do jogador no decorrer	do			
projeto	. 92			
Figura 22 – Fórum do jogo	. 93			
Figura 23 – Tela de Criar Projeto	. 94			
Figura 24 – Tela do módulo para criar Etapa	. 95			
Figura 25 – Funcionalidade para alocar jogadores ao projeto	. 95			
Figura 26 – Tela de configuração das aquisições	. 96			
Figura 27 – Turma de validação. IFRN - Campus Ipanguaçu. 10/01/2014	101			
Figura 28 – Turma de validação. IFRN - Campus Ipanguaçu. 10/01/2014	101			
Figura 29 – Turma de validação. UFERSA. 13/01/2014	102			

LISTA DE TABELAS

Tabela 1 – Exemplos de riscos comuns em projetos de software	16
Tabela 2 – Regras do sistema fuzzy para controle do afastamento por licença	70
Tabela 3 – Cenário da validação	73
Tabela 4 – Sexo e Faixa etária dos avaliadores	73
Tabela 5 – Resultado do questionário sobre a experiência prévia do jogador	74
Tabela 6 – Resultado do questionário respondido para avaliação do jogo	75

LISTA DE ABREVIATURAS

ABP – Aprendizagem Baseada em Projetos

AG - Algoritmo Genético

AI – Agentes Inteligentes

BC - Base de Conhecimento

BD - Base de Dados

BR – Base de Regras

CE – Computação Evolutiva

CF – Computação Flexível

FCM - Fuzzy C-Means

IA – Inteligência Artificial

ITU – União Internacional de Telecomunicações

JETS – Jogo da Equipe de Teste de Software

JPBN – Jogo Persistente Baseado em Navegador

MI – Motor de Inferência

PG - Programação Genética

SF – Sistema Fuzzy

SFBR - Sistema Fuzzy Baseado em Regras

SFG – Sistema Fuzzy Genético

SFGBR – Sistema Fuzzy Genético Baseado em Regras

SMA - Sistema Multiagente

SUMÁRIO

1 INTRODUÇÃO	10
1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO	10
1.2 PROPOSTA	12
1.3 ORGANIZAÇÃO	13
2 GESTÃO DE PROJETOS DE SOFTWARES	15
2.1 ABORDAGENS GERAIS	17
2.2 GESTÃO DE RISCOS EM PROJETOS	18
3 APRENDIZAGEM BASEADA EM PROJETOS, JOGOS E AG	ENTES
INTELIGENTES	21
3.1 APRENDIZAGEM BASEADA EM PROJETOS	21
3.1.1 Definições	21
3.1.2 Aprendizagem baseada em projetos versus abordagem tradicional .	22
3.2 TREINAMENTO BASEADO EM JOGOS	23
3.3 JOGOS	25
3.3.1 Jogos para navegador	25
3.3.2 Jogos persistentes baseados em navegador	27
3.3.3 Jogos educacionais	27
3.4 JOGOS SÉRIOS	29
3.5 AGENTES INTELIGENTES	30
4 SISTEMAS FUZZY	32
4.1 CONJUNTOS FUZZY	33
4.1.1 Funções de pertinência	34
4.1.2 Definições sobre conjuntos Fuzzy	37
4.2 COMPUTAÇÃO COM REGRAS FUZZY	38
4.2.1 Variáveis lingüísticas	38
4.2.2 Sintaxe das regras Fuzzy	40
4.2.3 Semântica das regras Fuzzy	42
4.2.4 Inferência com regras Fuzzy	45
4.3 SISTEMAS FUZZY BASEADOS EM REGRAS	46
4.3.1 Sistemas Fuzzy de classificação	47
4.3.2 Métodos de inferência	47
4.3.2.1 Método de Raciocínio Fuzzy Clássico	48

4.3.2.2 Método de Raciocínio Fuzzy Geral	49
5 REVISÃO BIBLIOGRÁFICA	51
5.1 JOGOS TRADICIONAIS E DE TABULEIRO PARA O ENSINO	DE
ENGENHARIA DE SOFTWARE	51
5.2 JOGOS DE COMPUTADOR PARA O ENSINO DE ENGENHARIA	DE
SOFTWARE	52
5.3 JOGOS DE COMPUTADOR PARA GESTÃO DE RISCOS	54
6 SUPERRISK	
6.1 CENÁRIO E ENREDO DA SIMULAÇÃO	56
6.2 VISÃO GERAL DO SUPERRISK	58
6.2.1 Funcionamento do SUPERRISK	
6.2.1.1. Administrador	60
6.2.1.2. Jogador	
6.3 DESCRIÇÃO DOS AGENTES	
6.3.1 Agente Dias	
6.3.2 Agente Nascimento	
6.3.3 Agente Das Chagas	64
6.3.4 Agente Sara	
6.3.5 Agente Costa	65
6.4 DETALHAMENTO DOS SISTEMAS INTELIGENTES	
6.4.1 Cálculo da pontuação do Jogador	66
6.4.2 Cálculo de percentual de Aumento de Salário	67
6.4.3 Sistema Fuzzy para afastamento dos funcionários	68
6.5 VALIDAÇÃO DO SUPERRISK	
6.5.1 Discussão sobre a avaliação dos alunos	
7 CONSIDERAÇÕES FINAIS	
7.1 PUBLICAÇÕES	77
7.2 TRABALHOS FUTUROS	
REFERÊNCIAS	
Anexo A - Detalhes Técnicos da Implementação do SUPERRISK	85
A.1 Visão Geral e Módulos do SUPERRISK	85
A.1.1 Módulo de Cadastro de Novo Jogador	
A.1.2 Módulo de Acompanhamento de Projeto	
A.1.3 Painel de pontuação dos jogadores	87

A.1.4 Módulo de Mercado de Trabalho88
A.1.5 Módulo de Equipe de Trabalho89
A.1.6 Módulo de Aquisições90
A.1.7 Módulo de Eventos91
A.1.8 Módulo de Fórum92
A.1.9 Módulo para Criação de Projetos93
A.1.10 Módulo para criar Etapas94
A.1.11 Módulo para alocar Jogadores95
A.1.12 Módulo de configurar Aquisições96
A.1.13 Módulo de acompanhar Jogadores96
Anexo B - Questionários 98
B.1 Primeiro Questionário98
B.2 Questionário Modificado
Anexo C – Registro da Validação101

1 INTRODUÇÃO

O ser humano aprende novas habilidades por meio do processo de tentativa e erro. Conceitos teóricos e abstratos geralmente são aprendidos, ou fixados, por meio da prática de atividades relacionadas aos mesmos.

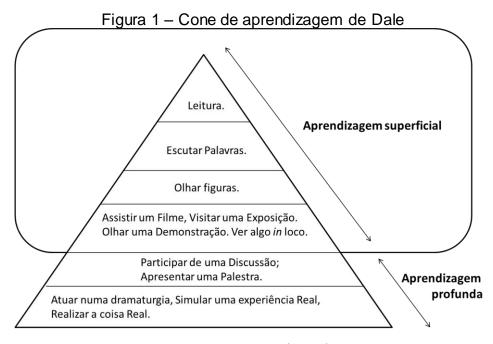
1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A adoção de métodos e princípios da gestão de projeto nas empresas de software são amplamente discutidas e aceitas. Nessas corporações, o principal papel do gerente é conduzir um projeto para uma conclusão bem sucedida. No entanto, isso não é o que geralmente acontece em relação a projetos de software: na prática, um elevado número de projetos falha devido a razões relacionadas à má gestão que, segundo alguns estudos, incluem a falta de certas habilidades dos gestores e deficiências dos métodos de ensino tradicionais (BREWER, 2005). De fato, a aprendizagem da gestão de projeto de software é uma tarefa que requer uma quantidade de treinamento prático que, por questões de tempo, dificilmente pode ser executado de forma satisfatória no decorrer de um curso de graduação, por exemplo.

Essas deficiências podem ser superadas com o uso de abordagens educativas complementares, tais como jogos e simulações. Visando atacar especificamente as deficiências no ensino e aprendizado da gestão de risco em projetos de softwares, esse trabalho apresenta o *SuperRisk*, uma ferramenta atraente, eficiente e colaborativa, baseada em jogos e na internet.

De fato, os jogos têm sido usados como opção para o aprendizado de diversas aptidões e para o treinamento em determinadas tarefas. As vantagens dos jogos no processo de aprendizagem incluem os baixos custos em comparação a treinamentos reais, o fato de não haver prejuízos ou riscos para o jogador em caso de erros, mesmo sendo possível assumir riscos, a possibilidade de repetir-se a mesma tarefa inúmeras vezes e a possibilidade de se trabalhar tarefas bastante específicas abstraindo-se o ambiente mais geral, entre outras. Dessa forma, é possível aprender a pilotar um avião (VINK, 2012), por exemplo, sem os custos envolvidos na tarefa real, ou de se aperfeiçoar técnicas de direção sem os perigos de se aprender nas ruas, por exemplo.

A Figura 1 mostra uma pirâmide com diversas atividades relacionadas à aprendizagem, divididas entre *aprendizagem superficial* e *aprendizagem profunda* de conteúdos.



Fonte: Wagner (1970)

Nesse modelo, definido por Wagner (1970) a simulação de experiências reais, incluindo os jogos, faz parte das atividades da aprendizagem mais profunda.

De fato, os métodos de ensino existentes no mundo contemporâneo são bastante abstratos, tendo em vista as incessantes aulas expositivas tornarem o aprendizado do aluno superficial. Isso se deve ao fato do foco educacional ser voltado para como o professor leciona e não para como o aluno deve absorver determinado conteúdo. Diante disso, nota-se que o perfil atual dos estudantes requer um ensino mais interativo, objetivando estimular a concentração dos alunos.

Com o desenvolvimento da computação, surgiram os jogos eletrônicos ou de computador que têm evoluído rapidamente. O primeiro jogo de computação foi desenvolvido Steve Russel em 1962 (LEVY, 1984), e simulava uma batalha entre duas naves espaciais representadas por simples setas em uma tela monocromática. Hoje temos jogos complexos que incluem técnicas avançadas de computação gráfica, inteligência artificial e iteração humano-computador para simular ambientes realísticos e permitir grande iteração com o jogador.

É possível encontrarmos na literatura diversas iniciativas para o ensino da computação que envolve jogos sérios usando diferentes abordagens desde o tradicional tabuleiro até o uso de jogos de computador. Essas abordagens vêm se popularizando com a grande necessidade de capacitação de profissionais em várias áreas da computação, dentre elas a de gestão de projetos de softwares.

Dentre as abordagens de jogos sérios encontradas na literatura, podemos citar os jogos tradicionais que usam tabuleiro ou cartas, como o jogo Lego SCRUM City (AGILE, 2013) e o Project Detective (von WANGENHEIM et al, 2011), ambos desenvolvidos para o ensino de metodologias de desenvolvimento de software.

Já entre os jogos para computador, podemos relacionar o TIM: *The Incredible Manager* (DANTAS et al, 2004), que é um jogo de simulação para um único jogador com foco no planejamento e controle do desenvolvimento de software, incluindo as tarifas de orçamento, cronograma e qualidade. O JETS (SILVA et al, 2011), que é um jogo multiusuário para o processo de ensino-aprendizagem de testes de software em cursos de graduação na área de computação; e o SimSE (NAVARRO; van DER HOEK, 2004), que foca no gerenciamento do projeto de software, englobando as fases de concepção, análise, projeto, construção e testes.

O jogo RSKManager (PALUDO, 2013) é um jogo de computador para apenas um usuário que aborda a identificação, a análise, o planejamento de respostas, o monitoramento e a execução de ações de tratamento de riscos em projetos de software. No jogo, os riscos são considerados como positivos ou negativos em relação aos objetivos do projeto de software abordado.

A proposta desse trabalho é parte de um projeto em andamento do Grupo de Pesquisa em Engenharia de Software (GPES)¹, que já produziu um jogo sério multiusuário baseado em navegador como um componente de apoio no processo de qualificação de novos profissionais em gestão de projetos (GALVÃO et al, 2012).

1.2 PROPOSTA

Esse trabalho propõe o uso de um jogo persistente, chamado *SuperRisk*, para o ensino de gestão de projetos de software baseado em navegadores para a internet. O *SuperRisk* engloba várias tarefas que um gerente de projeto deve

.

¹ http://www.les.ufersa.edu.br

executar em uma organização. Com foco na gestão de riscos, mais especificamente nas tarefas de planejamento, controle e monitoramento, incluindo o orçamento e cronograma. O *SuperRisk* usa Aprendizagem Baseada em Projetos (ABP) (AYAS; ZENIUK, 2003) para possibilitar ao jogador a aquisição de conhecimento em gerenciamento de projetos, especialmente no controle dos riscos envolvidos neste processo.

Foram usadas técnicas de agentes inteligentes de software no monitoramento e controle do ambiente para alterar, de forma autônoma, as variáveis do sistema que controlam os profissionais, as organizações e seus clientes do jogo. Outro diferencial do *SuperRisk* é o uso de um sistema fuzzy baseado em regras para possibilitar a inserção de conhecimento específico da área de gestão de risco no sistema.

O SuperRisk simula um ambiente complexo que permite a obtenção de gestão de projetos de softwares. Nele, cada jogador é capaz de gerir seus próprios gastos, estabelecendo metas e prazos a serem cumpridos. De fato, o jogador assume o papel de administrador da organização, responsável por prever os riscos e a produtividade da empresa. No SuperRisk, conceitos em relação a projetos de software são apresentados ao jogador de maneira natural, gradual e inconsciente, reduzindo os problemas advindos da execução dos projetos, ampliando-se a eficiência do processo, com o objetivo de minimizar custos e prejuízos.

O *SuperRisk* permite que o jogador avalie seu desempenho por meio de estatísticas. As ações efetuadas pelos jogadores são registradas e exibidas graficamente, facilitando a modelagem e planejamento de novas equipes de trabalho. De fato, o jogador pode aprender a partir de tentativas e erros gerando resultados cada vez melhores.

Outra vantagem do *SuperRisk* é a possibilidade de o jogador poder acessá-lo fora da sala de aula, maximizando sua experiência e aprendizado. O *SuperRisk* também permite que o professor ajuste o tempo de cada projeto de acordo com sua disponibilidade de tempo. Assim, um projeto pode ser definido com prazo máximo de algumas horas até semanas ou meses.

O SuperRisk foi testado e validado por 21 alunos durante 2 semanas na Universidade Federal Rural do Semi-Árido (UFERSA) e no Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte (IFRN). Os alunos avaliaram muito

positivamente o *SuperRisk*. De fato, os alunos que participaram dos testes e validação do *SuperRisk* pretendem indicar o uso do jogo aos colegas.

1.3 ORGANIZAÇÃO

O restante dessa dissertação está dividido da seguinte forma: No Capítulo 2 são apresentados os conceitos sobre gestão de projetos de softwares. O Capítulo 3 aborda a aprendizagem baseada em projeto na engenharia de software e o treinamento baseado em jogos de navegadores, além de conceitos sobre sistemas multiagentes. No Capítulo 4 são descritos conceitos sobre a lógica fuzzy e sistemas fuzzy para classificação. O Capítulo 5 apresenta uma revisão bibliográfica. O Capítulo 6 apresenta o *SuperRisk*, seguido pelas conclusões no Capítulo 7.

2 GESTÃO DE PROJETOS DE SOFTWARES

As empresas de desenvolvimento de software têm crescido em importância econômica devido à disseminação de computadores nos diversos setores da economia. Essa expansão tem gerado uma escassez de profissionais qualificados. Assim, para assegurar que o desenvolvimento e a manutenção de software sejam sistemáticos, disciplinados e qualificados, tais organizações têm adotado a aplicação de gerenciamento, planejamento, coordenação, mensuração, monitoramento e controle das atividades envolvidas na produção de software.

Desta forma, gerenciar um projeto de software inclui a tarefa de tentar prever os riscos que podem afetar o cronograma do projeto e/ou a qualidade do software que está sendo desenvolvido e tomar providências para evitar ou atenuar os impactos decorrentes desses riscos (HALL, 1998; OULD, 1999). Pode-se definir um risco como um evento indesejado com consequências negativas (ATLEE, 2009).

De acordo com a PMI (*Project Management Institute*), uma associação sem fins lucrativos para profissionais de gerenciamento de projetos (PMI, 2004), um risco em projetos é definido como sendo um evento ou condição incerta sobre um objetivo do projeto, como por exemplo o tempo, custo, escopo ou a qualidade. Os projetos de software estão expostos a incertezas das mais variadas origens e a gerencia de riscos é a área de estudo que enfoca o planejamento e o acompanhamento dessas incertezas (SCHWALBE, 2010).

Os projetos de softwares apresentam uma série de particularidades em relação a outros tipos de projetos, incluindo o seguinte:

- Os ambientes de aplicação envolvem diversas tecnologias;
- Há muita incerteza nos campos de atuação;
- Em diversas atividades n\u00e3o existem precedentes em que se possa fundamentar certas decis\u00f3es.

Os diversos riscos em um projeto de software podem ser divididos em três categorias (SOMMERVILLE, 2006):

- 1. Riscos de projeto: afetam o cronograma ou os recursos do projeto;
- Riscos de produto: afetam a qualidade ou o desempenho do software que está sendo desenvolvido;

 Riscos de negócio: afetam a organização que desenvolve ou adquire o software.

Dessa forma, é possível identificar quais são as implicações de um risco particular nos projetos e planejar como lidar com esses riscos, caso ocorram. A Tabela 1 traz alguns exemplos de riscos comuns em projetos de software.

Tabela 1 – Exemplos de riscos comuns em projetos de software.

Risco	Tipo	Descrição
Rotatividade de pessoal	Projeto	Profissionais com experiência, grande talento ou adequação ao projeto podem ser realocados antes do término, ou não estarem presentes em pontos críticos do desenvolvimento.
Mudança de gerência	Projeto	Reestruturação na gerência da organização pode inviabilizar a continuidade do projeto por mudança nas prioridades.
Baixo desempenho das ferramentas utilizadas	Produto	As ferramentas utilizadas que apoiam o projeto podem não funcionar conforme previsto.
Desenvolvimento de uma interface com usuário inadequada	Produto	A interface desenvolvida pode ter baixa aceitabilidade ou não ser de fácil usabilidade.
Concorrência de produto	Negócio	Um produto concorrente é lançado no mercado antes da conclusão do sistema.

Fonte: (SOMMERVILLE, 2006)

Os passos para se evitar que tais riscos comprometam um projeto, podem ser classificados de acordo com as quatro etapas, listadas a seguir (SOMMERVILLE, 2006):

- Identificação de riscos;
- Análise de riscos: previsão da probabilidade de acontecerem e das consequências que podem causar;
- Planejamento de riscos: elaboração de planos para tratar os riscos, seja evitando-os ou minimizando seus efeitos no projeto; e
- Monitoramento de riscos: Durante essa etapa os riscos são constantemente avaliados e os planos de mitigação dos riscos são revistos à medida que mais informações sobre eles são coletadas.

Os impactos e probabilidades de ocorrência de cada risco devem ser analisados para que as estratégias mais apropriadas sejam seguidas na tentativa de controlá-los e gerenciá-los. É importante salientar que muitos riscos não podem ser evitados. Assim, é preciso preparar um plano de contingência para os mesmos.

Nota-se que tais atividades relacionadas à gestão de projetos de software são geralmente ensinadas como parte dos cursos de engenharia de software.

Assim, o tempo para o aprendizado de todas as habilidades relacionadas à gestão de projetos de software é bastante limitado e, devido ao caráter prático das tarefas envolvidas na gestão de projetos de softwares que requerem tempo considerável para serem executadas, geralmente essas tarefas são apenas apresentadas de forma teórica, às vezes incluindo exemplos reais e/ou práticos.

2.1 ABORDAGENS GERAIS

No âmbito da Gerência de Riscos de projetos, nota-se a inclusão de processos envolvidos na identificação, análise e resposta aos riscos dos mesmos. Esse direcionamento inclui o ato de maximizar os resultados advindos de eventos positivos e minimizar aqueles advindos de eventos negativos.

Sendo assim, tal gerência agrega alguns grupos de processos. Dentre eles podemos citar o planejamento da própria gerência, a identificação dos riscos, a análise desses riscos de maneira qualitativa, a análise de maneira quantitativa, o planejamento das respostas aos riscos e o controle e a monitoramento de tais riscos - procurando identificar novos riscos. Corroborando, assim, para uma sistematização do trabalho a ser realizado.

Diversos projetos grandes de softwares sofreram fracassos por longas datas. Essa foi uma das primeiras indicações das dificuldades de gerenciamento dos projetos de softwares. Foi constatado que a falha residia na abordagem de gerenciamento utilizada: algumas técnicas aplicáveis de gerenciamento eram ineficazes para o desenvolvimento do software (SOMMERVILLE, 2006).

Diante do exposto, em (MACHADO et al., 2001) são citadas algumas práticas que compõem a Gerência de Riscos, as quais possibilitam o desenvolvimento do projeto. Dentre tais práticas podemos encontrar:

 Estabelecimento do escopo da gerência de risco, de acordo com as políticas organizacionais;

- Identificação dos riscos por todo o período do projeto;
- Análise e prioridade dos riscos, visto que os riscos devem ser analisados para priorizar a aplicação dos recursos com o objetivo de reduzir tais riscos;
- Definição de estratégias para favorecer uma gerência ideal dos riscos em nível de projeto ou organização;
- Execução das estratégias definidas para a gerência de risco;
- Avaliação dos resultados da estratégia da gerência de risco;
- Execução das ações corretivas para corrigir ou evitar um impacto do risco, caso o progresso esperado na redução de tal risco não seja alcançado.

Segundo (PAULA FILHO, 2003) os riscos devem ser estimados, sendo essa uma atividade pouco praticada pelas organizações. Isso porque um bom planejamento prevê o que pode acontecer se tudo ocorrer bem, e também se tudo ocorrer mal, isto é, as consequências positivas e negativas.

Vale salientar que os riscos não são constantes durante todo o processo: alguns podem desaparecer e surgir no decorrer do processo, alterando, com isso, as probabilidades de impactos e, por conseguinte, mudando a prioridade.

2.2 GESTÃO DE RISCOS EM PROJETOS

A gestão de riscos é imprescindível para projetos de softwares, sendo composta por atividades coordenadas para direcionar uma organização em relação ao risco. Assim, os gerentes de projetos deveriam avaliar regularmente os riscos durante o processo de desenvolvimento para minimizar as chances de fracassos, contribuindo, portanto, para um progresso eficiente (PIVETTA, 2002).

Observa-se que as empresas sentem grande dificuldade de introduzir os conceitos da gestão de riscos no ambiente de trabalho. Um exemplo disso são os dados apresentados pelo relatório do *PMSurvey* (CHAPTERS, 2013), onde em 59% das organizações, o tratamento de riscos é realizado informalmente, conforme o interesse ou necessidade do responsável pelo projeto.

Os objetivos principais da gestão de riscos são a prevenção e a mitigação de tais riscos. Eles são formulados de maneira realista, tornando o projeto mais previsível e controlável. Sendo assim, os problemas de cronograma, orçamento e funcionalidade dos softwares podem ser controlados através de ações preventivas.

Desse modo, para que tudo ocorra como planejado, a gestão de riscos envolve seis atividades, descritas a seguir:

- Identificação dos Riscos: antes que os riscos se tornem problemas, é
 primordial identificá-los, objetivando um levantamento inicial de todas
 as possibilidades de riscos existentes;
- Análise de Riscos: Essa atividade serve para a caracterização dos riscos, tendo como objetivo explorar as melhores estratégias de mitigação. Assim, após a análise dos riscos, uma lista classificada pela importância dos riscos deve ser elaborada, priorizando os esforços de atuação;
- Planejamento dos Riscos: envolve a determinação dos riscos a serem gerenciados, visto que os planos de ação necessitam estar sob controle, assim como os plano de contingência para os riscos que se encontram além da capacidade de mitigação;
- Monitoração dos Riscos: este processo consiste em observar o status dos riscos e as ações tomadas para evitá-lo de forma a favorecer a gerência para que esta atue de modo preventivo e não reativo aos eventos. No entanto, a frequência de atualização do processo depende dos recursos disponíveis e da rapidez com que o projeto se desenvolve.
- Controle dos Riscos: consiste na avaliação da situação corrente para determinar eventuais desvios do planejado. Observe que quando um risco tem suas atividades encerradas, todas as informações relacionadas devem ser documentadas.
- Comunicação dos Riscos: Comunicação entre os membros do projeto de software é um dos principais fatores para uma realização bem sucedida da gestão de riscos. Sem uma comunicação exemplar, o gerenciamento se torna inviável, aparecendo riscos, problemas e crises para a organização.

A partir da análise das ideias citadas acima, nota-se que, como afirmou (CHADBOURNE, 1999) os riscos, em software, não podem ser meros tópicos da agenda, devem ser o "coração" do negócio, como ocorre nas demais áreas. Sendo assim, a gerência de riscos é um processo de software, o qual busca averiguar os fatores que favorecem aos riscos com o objetivo de reduzir seus efeitos, favorecendo a elaboração de um produto com qualidade, que atenda as necessidades dos usuários e, principalmente, se enquadre no prazo e esteja de acordo com os custos estimados.

Para (SOMMERVILLE, 2006) os tipos de risco que afetam um projeto dependem tanto do projeto quanto do ambiente organizacional em que o software esta sendo desenvolvido. Contudo, muitos riscos são considerados universais como, por exemplo: a rotatividade de pessoal, caracterizado quando os funcionários mais experientes tendem a deixar o projeto antes do término, e os atrasos na especificação que ocorre quando as especificações essenciais não estão disponíveis dentro do prazo definido.

O CMMI (Capability Maturity Model Integration) é um conjunto de práticas relacionadas que quando implementadas coletivamente satisfazem diversos objetivos considerados importantes para aprimorar, avaliar e melhorar o processo de desenvolvimento de software. O CMMI é composto por 22 áreas de processo e uma dessas é a RSKM (Gerenciamento de risco), no terceiro nível de maturidade - Definido, que propõe um conjunto de ações que devem ser realizadas continuamente durante o ciclo de vida do projeto com o objetivo de minimizar proativamente o impacto dos riscos no projeto.

No próximo capítulo, são apresentados conceitos sobre Aprendizagem Baseada em Projetos, Jogos e Agentes Inteligentes.

3 APRENDIZAGEM BASEADA EM PROJETOS, JOGOS E AGENTES INTELIGENTES

Esse capítulo apresenta e discute os conceitos principais relacionados à aprendizagem baseada em projetos, o treinamento baseado em jogos, jogos para navegador e agentes inteligentes, com o foco no uso conjunto de tais técnicas para o ensino de Gestão de Projetos.

3.1 APRENDIZAGEM BASEADA EM PROJETOS

A Aprendizagem Baseada em Projetos (ABP) (*Project-Based Learning* - PBL em inglês) foi proposta por Ayas; Zeniuk (2003) como um meio de conduzir o ensino onde as competências pessoais e coletivas são construídas através do desenvolvimento de capacidades de aprendizagem que permitem o raciocínio em curto prazo. Esta teoria de aprendizagem possibilita a criação e o compartilhamento do conhecimento para além do individual e da equipe. Esta teoria pressupõe que, em um projeto centrado em tarefas onde as pressões de curto prazo prevalecem, os participantes devem equilibrar a ação e a reflexão a fim de construir a competência da capacidade de aprendizagem (HSU; LIU, 2005).

3.1.1 Definições

A ABP pode ser entendida como uma técnica de ensino onde situações de um contexto real são modeladas em um projeto fictício no qual os alunos devem empenhar-se para concluí-lo, podendo assim construir conhecimentos a respeito daquela experiência.

Um *projeto* é definido como um esforço temporário, conduzido por um grupo de pessoas, para criar um produto, serviço ou resultado único. *Temporário* significa que todos os projetos possuem início e fim definidos. O fim é atingido quando os objetivos são alcançados, caso contrário, o projeto fracassou.

Um *projeto* é caracterizado por seus objetivos e requisitos claramente definidos (*escopo*) e obedece a um orçamento aprovado (*custo*). Além disso, o projeto deve ser finalizado no prazo especificado (*tempo*) (SCHWABER; BEEDLE, 2001).

Na ABP, o aprendizado acontece por meio de um processo no qual os alunos, trabalhando em equipe, constroem um produto. O *produto* pode ser algo tangível (como um modelo ou protótipo, um sistema ou um robô), um produto informático (tais como software, uma apresentação ou um produto multimídia), ou um produto escrito (tal como um relatório, uma avaliação ou um resumo dos resultados experimentais). O produto deve responder a uma pergunta e resolver um problema além de atender às exigências e necessidades definidas pelo instrutor do curso ou identificadas pelos alunos.

3.1.2 Aprendizagem baseada em projetos versus abordagem tradicional

A ideia central da ABP é que os problemas do mundo real captam o interesse dos alunos e provocam reflexões sérias sobre como os alunos podem adquirir e aplicar novos conhecimentos em um contexto de resolução de problemas (XIAHOU, 2010). Nesse contexto, o professor desempenha o papel de facilitador, trabalhando com os alunos para formular perguntas relevantes, estruturar tarefas significativas, treinando tanto o desenvolvimento de conhecimentos e habilidades sociais, como também avaliando cuidadosamente o que os alunos aprenderam com a experiência. Os defensores afirmam que a ABP ajuda os alunos a adquirirem as habilidades de pensamento e colaboração necessárias no local de trabalho.

A ABP está focada em questões que levam os alunos a encontrar os conceitos fundamentais e os princípios de um assunto por meio da prática. A PBL auxilia as habilidades de pensamento criativo, mostrando que existem muitas maneiras de resolver um problema. Além disso, o aprendizado baseado em projetos difere da investigação tradicional por sua ênfase na construção pelos próprios artefatos dos alunos para representar o que está sendo aprendido.

A abordagem tradicional no ensino de engenharia de software é baseada em um modelo de aulas teóricas e expositivas de leitura. Porém, devido ao caráter prático deste modelo, essa abordagem dificulta o aprendizado dos alunos, pois o pouco envolvimento com o tema dá a eles um papel passivo, tornando-os apenas espectadores. Diferentemente do papel do engenheiro de software, que deve estar atento ao que ocorre nos projetos que coordena e tomar decisões que são as bases para o término bem sucedido dos projetos.

Na abordagem tradicional, os problemas normalmente são adaptados e simplificados de tal forma que não parecem ser relevantes ou estão ligados a soluções já pré-fabricadas. Assim, os alunos não são estimulados a raciocinarem e criarem suas próprias ideias para lidar com os problemas apresentados. Além disso, alguns temas mais teóricos, como modelos de processo, parecem ser tão abstratos que não despertam nos alunos a ideia de como são usados na prática.

Ao integrar a ABP à aprendizagem de engenharia de software, o aluno pode obter um ganho de experiência que provavelmente não pode ser obtido nas atividades convencionais da sala de aula.

Assim, é possível notar que a ABP aplicada a softwares para ensino como, por exemplo, relacionados à engenharia de software é viável e útil aos alunos no tocante ao treinamento prático dos conhecimentos que devem ser transmitidos pela disciplina. Esta junção pode trazer também uma melhoria na capacidade analítica dos alunos, característica profissional que diz respeito à habilidade de analisar dados, nem sempre relacionados entre si, e, a partir desta análise, produzir informação de valor ou conhecimento.

3.2 TREINAMENTO BASEADO EM JOGOS

Com a evolução constante da humanidade, muitos paradigmas têm sido quebrados e muito se tem ganhado com isso. No âmbito da educação, o que se tem visto são novas metodologias e abordagens educacionais sendo criadas e recriadas com o intuito de alcançar melhores resultados pedagógicos com um público cada vez mais dinâmico e diversificado. Uma dessas abordagens é o uso de jogos eletrônicos no contexto educacional (WILSON, 2011), que vem sendo globalmente discutida e aplicada na educação formal e não-formal.

A educação formal representa um conjunto de atividades educacionais com objetivos claros e bem definidos, que obedecem a diretrizes educacionais centralizadas e que são executadas nos limites das escolas e universidades.

Já a educação não formal é mais difusa, menos hierárquica, desburocratizada e normalmente executada fora dos limites do sistema formal (escolas, universidades, etc.).

Alguns autores consideram também a existência da educação informal que, embora não apresente o rigor e a aplicação de metodologias pedagógicas

presentes nas demais, representa meios de aquisição de capacidades e conhecimentos em relação a alguma atividade (SORENSEN; MEYER, 2007).

Em meio aos diversos modelos educacionais, alguns fatores, como falta de motivação ou de envolvimento com o tema, muitas vezes tornam-se obstáculos bastante difíceis de serem transpostos e podem comprometer o rendimento do aluno. Neste sentido, os jogos podem ser integrados aos modelos de treinamento, adicionando elementos lúdicos, efeitos visuais e uma forma de interação mais atraente para os estudantes.

Alguns dos principais elementos encontrados em jogos incluem: interação, flexibilidade, competição, feedback visual e efeitos dramáticos (PRENSKY, 2001). Esses elementos são descritos a seguir:

- A interatividade permite ao jogador uma participação ativa no processo, evitando que o mesmo se entedie e perca o interesse.
- A flexibilidade permite ao jogador um maior nível de controle e investigação do conteúdo, seguindo seu próprio caminho, suas decisões e tempo, porém, de acordo com regras bem definidas.
- A competição representa o desafio, o conflito, a solução de problemas que requerem exploração e habilidades estratégicas que motivam o jogador na busca dos objetivos definidos.
- O jogo apresenta o progresso do jogador na busca dos seus objetivos por meio de *feedback* e avaliação. Este *feedback* é feito por meio de efeitos dramáticos sobre os elementos que compõem uma história para o jogo.
- Estes efeitos (sons, imagens, animações, músicas, humor e histórias)
 relacionam entretenimento ao conteúdo do aprendizado, adicionando
 um contexto, personagens e dramatização, permitindo que o estudante
 tenha a chance de vivenciar um papel e atuar dentro do jogo.

Ao usar jogos multi-usuários de computador ou tradicionais para fins educativos, diversos aspectos do processo de aprendizagem devem ser observados. Por exemplo:

- Os jogadores, na figura de alunos devem ser estimulados a combinar os conhecimentos de diferentes áreas para escolher uma solução ou para tomar uma decisão em um determinado ponto;
- II. Os alunos devem ser capazes de testar como o resultado do jogo muda com base em suas decisões e ações;
- III. Os alunos devem ser encorajados a manter contato com outros membros da equipe para discutir e negociar as etapas subsequentes, melhorando, assim, entre outras coisas, suas habilidades sociais (PIVEC t al., 2003).

Os alunos de engenharia de software, em especial, podem se beneficiar com o uso de jogos no treinamento, uma vez que o gerenciamento de projetos depende fortemente da experiência adquirida na prática. O jogo proporciona a oportunidade de analisar situações já vivenciadas e avaliar diferentes caminhos alternativos para o projeto de acordo com decisões específicas a serem tomadas em pontos particulares do jogo, melhorando assim as habilidades de gerenciamento e tomada de decisões do jogador.

3.3 JOGOS

Os jogos possuem a competência motivacional para prover ao jogador a capacidade de superar os desafios, adversários e até a si mesmo. Dessa forma ela é caracterizada como uma atividade que além de divertida, é por vezes viciante sendo um grande foco de atenção para os jogadores. A seguir alguns tipos de jogos serão apresentados.

3.3.1 Jogos para navegador

De acordo com pesquisa realizada pela União Internacional de Telecomunicações (UIT) (ITU, 2010), de 2003 para 2009 o número de pessoas com acesso à internet naquele período representava 25% da população mundial. Em países desenvolvidos esse número sobe para 64%. Seguindo esse crescimento, de 2009 para 2010, o número de usuários que acessam a Internet em casa passou de 1,4 para 1,6 bilhão (SARKAR, 2014).

Face ao crescimento do público "conectado", a plataforma para internet - plataforma na qual os jogos podem ser rodados na própria internet - se torna cada vez mais atrativa. Aliado a esse crescimento, é possível observar as mudanças sociais ocorridas nos últimos 25 anos, das quais surgiram o que se conhece hoje como "nativos digitais" (PRENSKY, 2001). Os nativos digitais são pessoas com contato fácil e permanente com a tecnologia, representando hoje cerca de 50\% da população ativa e podendo chegar a 80% em 10 anos (MONTEIRO, 2011).

Visando essa fatia de mercado em constante crescimento, a bilionária indústria de jogos eletrônicos começou a investir em títulos 100% Web. Esse investimento da indústria de jogos é focado nas potencialidades que a plataforma da internet traz consigo, tais como:

- Persistência: Manutenção do estado do jogo no servidor para posterior continuação. Ou seja, o jogador pode parar/pausar o jogo a qualquer momento e retorná-lo quando quiser ou estiver disponível;
- Independência de plataforma: Requisito cada vez mais necessário face à diversificação de sistemas operacionais e dispositivos de acesso à internet, tendo em vista os muitos sistemas operacionais usados em computadores e dispositivos móveis;
- Colaboração vivenciada na Web 2.0;
- Explosão das redes sociais.

Os jogos baseados em navegador podem incluir todos os gêneros de jogos e podem ser para apenas um jogador (*single-player*) ou para vários (*multiplayer*). Os jogos *multi-usuário* têm recebido mais destaque devido ao seu enfoque adicional sobre a interação social, muitas vezes em escala maciça. As redes sociais fazem uso destes jogos para aproximar mais pessoas e maximizar lucros. Um exemplo desse uso é o jogo chamado Farmville da empresa Zynga (ZYNGA, 2011). O jogo Farmville é baseado em navegador e simula uma fazenda virtual onde aproximadamente 40 milhões de pessoas jogam Farmville mensalmente.

3.3.2 Jogos persistentes baseados em navegador

Um Jogo Persistente Baseado em Navegador (*Persistent Browser-Based Game* - PBBG) é um jogo eletrônico que pode ser jogado e acessado em um navegador da Web e que apresenta um ambiente virtual compartilhado persistente. Assim, nesse ambiente os eventos continuam a acontecer mesmo na ausência do jogador conectado, podendo este recuperar sua sessão posteriormente (PMI, 2004).

Em meio à grande diversidade de públicos, com tempo e meios cada vez mais específicos, estas características proporcionadas pelos PBBGs podem ser um grande diferencial no sucesso ou não de uma ferramenta. De fato, os PBBGs permitem que pessoas com disposições distintas possam interagir, competir ou trocar experiências antes impossibilitadas por estas limitações. As pessoas que necessitam se locomover podem ser beneficiadas com a portabilidade dos PBBGs, uma vez que as informações sobre os perfis dos usuários são armazenadas do lado do servidor, e nenhum software especial é necessário.

Assim os jogos podem ser acessados em qualquer terminal através de um navegador e os jogadores podem continuar suas sessões de onde pararam.

Os usuários de dispositivos móveis, celulares, *smartphones* e *tablets*, por exemplo, muitos destes com capacidade limitada, podem interagir, já que boa parte dos jogos de navegadores são leves, pois não usam gráficos complexos, sons ou imagens de alta definição. Uma relação vasta e detalhada dos principais jogos de navegadores disponíveis no momento pode ser encontrada em (OCIGRUP, 2011).

3.3.3 Jogos educacionais

Jogos educacionais são simulações que levam os aprendizes a praticarem tarefas de alta interatividade, possibilitando lhes o desenvolvimento de habilidades sem risco e sem custo, através do uso do computador (TEIXEIRA, 2007).

Para Allué (1999), jogos educacionais constituem por qualquer atividade de formato instrucional ou de aprendizagem que envolva competição e que seja regulada por regras e restrições.

Os jogos podem ser ferramentas de ensino bastante eficientes, pois além de divertidos eles são motivadores, dessa forma a criatividade e vontade de competir das pessoas são ampliadas, facilitando o aprendizado.

De acordo Prensky (2001) para que os jogos possam realmente ser determinados como jogos, eles devem possuir seis características fundamentais, são elas:

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

Segundo Alves (2010) quando há a necessidade de aprender conceitos por meio da experiência, os jogos expressam características de simuladores que procuram reproduzir situações da vida real que não podem ser vivenciadas diretamente, como acontece no planejamento de projetos de software.

Os jogos educacionais tornaram-se uma possibilidade atraente para essa conjuntura, visto que permitem complementar o ensino de conceitos básicos através da demonstração da aplicação de tópicos relevantes. Especificamente, os jogos baseados em computador admitem a simulação de episódios reais e o fornecimento de *feedback* imediato, estimulando os estudantes a moldarem seu aprendizado por meio de acertos e erros e, principalmente, da análise das suas decisões.

3.4 JOGOS SÉRIOS

Apesar da maior parte dos jogos serem desenhados para entretenimento, há bastantes que são criados com outros propósitos. Estes podem ser catalogados pela sua finalidade ou pelo conteúdo que apresentam.

Um Serious Game, ou Jogo Sério, é desenhado com um único objetivo: melhorar algum aspecto da aprendizagem. Ao contrário dos outros tipos de jogos, pretende-se transmitir conteúdos educativos ao jogador, deixando o entretenimento para segundo plano. Consiste num ambiente simulado, onde o jogador é o interveniente principal.

Pretende-se que este execute tarefas, processos e tire conclusões, de forma a aumentar a sua produtividade e os seus conhecimentos, sem correr riscos ou sem desencadear na vida real uma sucessão de eventos que seriam negativos para o negócio.

Algumas vantagens são enumeradas por (McGONIGAL, 2013) para a utilização de jogos na aprendizagem, destacam-se as mais relevantes:

- 1. Desenvolver capacidades intelectuais e motoras;
- 2. Perceber a aplicabilidade das matérias abordadas;
- 3. Motivar e aumentar a atenção do jogador;
- 4. Aumentar o relacionamento e colaboração entre indivíduos;
- 5. Tornar o jogador mais optimista e criativo na solução de problemas;
- 6. Permitir uma evolução em fatores chaves da produtividade;
- 7. Criar uma forte vontade de superar as dificuldades e problemas que surjam;
- 8. Permitir criar experiência numa área de formação;
- 9. Aumentar a confiança nas suas capacidades;
- 10. Criar um sentido de segurança e de realização ao evoluir no jogo;
- 11. Permitir que o jogador seja mais ousado na procura de soluções;
- 12. Criar um sentido de rivalidade saudável entre jogadores;
- 13. Reforçar relações pessoais.

3.5 AGENTES INTELIGENTES

Visando melhorar a eficácia, ou até mesmo a autonomia das ferramentas computacionais, algumas técnicas de Inteligência Artificial (IA) vêm sendo empregadas nas mais diversas áreas.

Devido a algumas de suas capacidades, tais como: i) comportamento guiado por objetivos, ii) reatividade, iii) raciocínio, iv) adaptabilidade, v) aprendizagem, vi) comunicação e cooperação, os Agentes Inteligentes (AI) têm ganhado espaço e se tornado populares em jogos de computador.

Um agente inteligente pode ser definido como qualquer entidade que possa receber informações do ambiente onde vive por meio de sensores e agir nesse ambiente por meio de atuadores, de maneira racional, isto é, de maneira correta e tendendo a maximizar um resultado esperado (RUSSELL; NOVING, 2004).

Existem vários tipos de agentes, sendo que cada um é elaborado de acordo com o ambiente no qual é inserido e a funcionalidade que deve prover. Eles podem ser de software ou de hardware e normalmente são classificados em 4 tipos básicos (RUSSELL; NOVING, 2004):

- 1. Agentes reativos simples: são aqueles que selecionam suas ações com base na percepção atual, ignorando o histórico das percepções;
- Agentes reativos baseados em modelo: mantêm o estado interno para controlar aspectos do mundo que não estão evidentes na percepção corrente;
- Agentes baseados em objetivos: necessitam de uma descrição do estado atual, além de alguma espécie de informação sobre objetivos que descrevam situações desejáveis, para que esses possam agir visando alcançar seus objetivos;
- 4. Agentes baseados na utilidade: utilizam uma medida de desempenho que permita uma comparação entre diferentes estados do mundo, para selecionar o que terá maior utilidade a eles.

Um conjunto de agentes autônomos que colaboram entre si objetivando a solução de um problema que está além da capacidade de um único agente é considerado um Sistema Multiagente (SMA) (PONTES et al., 2010).

À medida que os jogos se tornam mais realistas, tanto em termos de física quanto de gráficos, também os personagens e o ambiente evoluem e tornam-se mais inteligentes. Os agentes são geralmente utilizados em jogos de computador para prover ao sistema comportamentos menos óbvios e diferentes a cada atuação, fazendo com que o jogador tenha que mudar sua estratégia e melhorar seu desempenho para poder superar as dificuldades apresentadas.

A lA está presente em todos os estilos de jogos, mais frequentemente em *Role Playing Games* - RPGs. Nesses jogos, os inimigos podem ser monstros ou personagens semelhantes ao jogador, controlados pelos agentes. Eles devem ser autônomos e precisam interagir com ambientes dinâmicos e complexos, o que requer comportamento reativo, planejamento e senso comum. Eles precisam navegar pelo extenso mundo virtual, necessitando buscar rotas e ter raciocínio espacial e raciocínio temporal. Os agentes avançados podem ter mecanismos para adaptarem-se às estratégias de seus oponentes, e podem inclusive aprender.

Os agentes inteligentes podem ser usados em jogos com inúmeros fins, não só na representação de oponentes ou parceiros, mas também do próprio ambiente em si, visto que os ambientes virtuais tentam representar, da forma mais fidedigna possível, os ambientes reais. Estes, por sua vez, estão sujeitos às mais diversas ocasionalidades, mudanças climáticas e/ou catástrofes, entre outras coisas que podem influenciar o andamento do jogo.

No contexto desse trabalho, são usados agentes inteligentes, com o objetivo de simular o ambiente de gestão de projetos e permitir que os alunos treinem de forma realística e motivadora.

No próximo capítulo são apresentados os fundamentos da teoria de conjuntos fuzzy, lógica fuzzy e sistemas fuzzy baseados em regras.

4 SISTEMAS FUZZY

Sistemas Fuzzy podem ser definidos como sistemas que utilizam a teoria de conjuntos fuzzy proposta por Lofti A. Zadeh em 1965 para representar pelo menos uma de suas variáveis (ZADEH, 1965). A teoria dos conjuntos fuzzy permite a representação e processamento de informações imprecisas e incertas, abundantes no mundo real, muitas vezes difíceis de serem processadas pelos computadores, e por isso é de grande utilidade para o projeto de sistemas que lidam com domínios que apresentam incerteza e imprecisão.

Os sistemas baseados em lógica fuzzy, ou Sistemas Fuzzy, usam um mecanismo de raciocínio baseado no raciocínio aproximado que possui grande habilidade para expressar a ambigüidade e subjetividade presentes no raciocínio humano (PEDRYCZ; GOMIDE, 1998). As bases de regras dos SF armazenam conhecimento representado, geralmente, por meio de regras do tipo:

Se → Então

Os antecedentes das regras relacionam as entradas do sistema, enquanto os consequentes relacionam as saídas, usando operadores lógicos. Um sistema de inferência, baseado em graus de pertinência e operadores de associação, é utilizado para a obtenção de uma saída a partir de dados de entrada.

Os SF têm sido aplicados com sucesso em várias áreas do conhecimento, desde controle, reconhecimento de padrões, modelagem, classificação e visão computacional até a vida artificial (ICHIBA et al., 2006; MAGDALENA; MONASTERIO, 1997; CHUNG et al., 2000; CHI et al., 1996; CINTRA et al., 2012; DE OLIVEIRA et al., 2013; TEIXEIRA JUNIOR et al., 2013; CINTRA; NEVES, 2013). Um número significativo de implementações vem consolidando os SF não só em instalações industriais, mas também em muitos produtos manufaturados de uso diário (PEDRYCZ; GOMIDE, 1998).

Nas próximas seções são discutidos os fundamentos da lógica fuzzy e dos sistemas fuzzy. Todas as definições da teoria de conjuntos fuzzy apresentadas neste capítulo podem ser encontradas de forma integral em (NICOLETTI; CAMARGO, 2004; PEDRYCZ, 1998) e (KLIR; YUAN, 1995).

4.1 CONJUNTOS FUZZY

A ideia de conjunto é bastante simples e comum na vida cotidiana e ocorre freqüentemente quando tentamos organizar ou sumarizar o conhecimento sobre objetos. Seja X um conjunto de objetos denominado *universo de discurso*, e x um elemento de X. Um conjunto A, $A \subseteq X$, é definido como uma coleção de objetos de $x \in X$.

Na teoria clássica dos conjuntos, chamados de conjuntos *crisp*, um elemento pertence ou não a um dado conjunto. Este fato pode ser expresso pela função apresentada em (1), chamada de *função característica*, que atribui o valor \$1\$ (um) ao elemento se ele for membro do conjunto, ou \$0\$ (zero), caso contrário.

$$F_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases} \tag{1}$$

Entretanto, o mundo real apresenta situações em que os conjuntos clássicos não são capazes de definir de forma adequada certas classes de elementos por não possuírem flexibilidade suficiente e forçarem uma transição abrupta entre a pertinência absoluta e a não pertinência absoluta.

Já os conjuntos fuzzy, considerados uma generalização dos conjuntos clássicos, possibilitam a transição gradual de significado, permitindo que cada elemento pertença parcialmente a um determinado conjunto.

Um conjunto fuzzy A é caracterizado por uma função de pertinência que associa os elementos do universo de discurso X a um valor em um determinado intervalo, geralmente [0,1]. Uma das representações mais usadas para denotar o grau de pertinência de um elemento x no conjunto A é A(x), ou seja:

$$A:x \rightarrow [0,1]$$

O grau de pertinência 0 (zero) significa a exclusão total de um elemento em relação ao conjunto e o grau de pertinência 1 (um) significa pertinência total do elemento ao conjunto. É possível representar um conjunto A como um conjunto de pares ordenados da forma A(x)/x como expresso abaixo:

$$A = \{ (A(x)/x \mid x \in X) \}$$

Como exemplo, considere $X = \{15, 18, 22, 28, 30, 33, 35, 39\}$ uma coleção de valores de temperaturas. O conjunto fuzzy definido por $A = \{(0.8/15), (0.9/18), (1/22), (0.7/28), (0.5/30), (0.3/33), (0.1/35), (0/39)\}$, pode ser entendido como o conjunto das temperaturas da categoria "média".

4.1.1 Funções de pertinência

A base da teoria dos conjuntos fuzzy é o conceito de pertinência. Em princípio, qualquer função da forma $A:x \to [0,1]$ descreve uma função de pertinência associada a um conjunto fuzzy A. As funções de pertinência mais utilizadas, porém, geralmente são representadas nos formatos triangular, trapezoidal, S ou gaussiano por terem representação simples e fácil utilização. A lém destes formatos tradicionais existe uma forma (particular do conjunto fuzzy) bastante utilizada em aplicações práticas: o conjunto unitário (singleton).

As funções mais conhecidas, bem como seus gráficos, são apresentados a seguir.

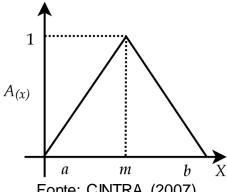
• Função Triangular: definida pelos parâmetros a, m e b, sendo $a \le m \le b$, como descrito abaixo:

$$A(x) = \begin{cases} 0 & \text{se } x \le a \\ \frac{x-a}{m-a} & \text{se } x \in (a,m) \\ 1 & \text{se } x = m \\ \frac{b-x}{b-m} & \text{se } x \in (m,b) \\ 0 & \text{se } x \ge b \end{cases}$$

• Função Trapezoidal: definida pelos parâmetros a, m, n e b, sendo $a \le m \le n \le b$, como descrito a seguir:

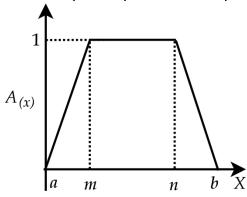
$$A(x) = \begin{cases} 0 & \text{se } x \le a \\ \frac{x-a}{b-a} & \text{se } x \in (a,m) \\ 1 & \text{se } x \in (a,m) \\ \frac{b-x}{b-n} & \text{se } x \in (n,b) \\ 0 & \text{se } x \ge b \end{cases}$$

Figura 2 – Função de pertinência triangular.



Fonte: CINTRA (2007)

Figura 3 - Função de pertinência trapezoidal



Fonte: CINTRA (2007)

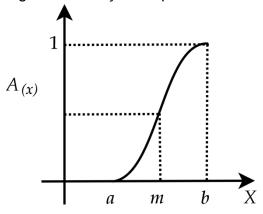
• Função S: definida pelos parâmetros a, m e b, como descrito a seguir. O ponto m=(a+b)/2 é conhecido como cruzamento da função S.

$$A(x) = \begin{cases} 0 & \text{se } x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2 & \text{se } x \in (a,m) \\ 1-2\left(\frac{x-b}{b-a}\right)^2 & \text{se } x \in (m,b) \\ 1 & \text{se } x > b \end{cases}$$

• Função Gaussiana: definida pelos parâmetros m e k, sendo k > 0, como descrito a seguir:

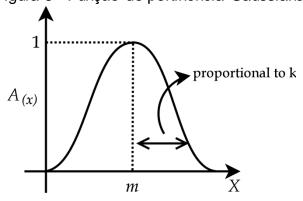
$$A(x) = e^{-k(x-m)^2}$$

Figura 4 - Função de pertinência S.



Fonte: CINTRA (2007)

Figura 5 - Função de pertinência Gaussiana.

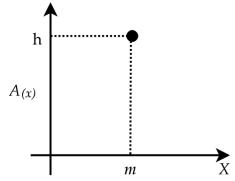


Fonte: CINTRA (2007)

• Conjunto Unitário (singleton): parâmetros (m, h), tal que $h \le 1$.

$$A(x) = \begin{cases} h & se \ x = m \\ 0 & se \ x \neq m \end{cases}$$

Figura 6 - Função de pertinência unitária.



Fonte: CINTRA (2007)

4.1.2 Definições sobre conjuntos Fuzzy

Alguns conceitos básicos estão associados aos conjuntos fuzzy. Entre eles estão os conceitos de $Corte-\alpha$, $Corte-\alpha$ Forte, Conjunto de Níveis, Suporte, Núcleo, Altura, Normalização e Conjunto Normal. Sejam A e B conjuntos fuzzy sobre um conjunto base X, também chamado de conjunto universo, e seja x um elemento do conjunto base, estes conceitos são apresentados a seguir.

 O Corte-α de um conjunto fuzzy A é o conjunto de todos os elementos do universo X com grau de pertinência em A maior ou igual a α, para α ∈ [0,1].

$$A_{\alpha} = \{x \in X \mid A(x) \ge \alpha\}$$

O Corte-α forte de A é o conjunto com todos os elementos do universo
 X com grau de pertinência maior que α, para α ∈ [0,1].

$$A_{\alpha F} = \{ x \in X \mid A(x) > \alpha \}$$

• O conjunto de níveis de A (η_A) é o conjunto dos valores dos graus de pertinência do conjunto A.

$$\eta_A = \{ \alpha \mid A(x) = \alpha, \forall x \in X \}$$

• O suporte de um conjunto fuzzy A S_A é o conjunto binário contendo todos os elementos de X com grau de pertinência diferente de 0 em A.

$$S_A = \{x \in X \mid A(x) > 0\}$$

• O núcleo de um conjunto fuzzy A N_A é o conjunto binário contendo todos os elementos de X com grau de pertinência igual a 1 em A.

$$N_A = \{x \in X \mid A(x) = 1\}$$

 A altura de um conjunto fuzzy A H_A é definida como o maior grau de pertinência obtido por qualquer elemento do conjunto A.

$$H_A = \sup_{x} \{A(x) \mid x \in X\}$$

Um conjunto fuzzy é considerado como conjunto normal se sua altura é
igual a 1. Caso contrário ele é chamado de subnormal. Os conjuntos
subnormais têm como núcleo um conjunto vazio.

4.2 COMPUTAÇÃO COM REGRAS FUZZY

Regras são úteis para a representação de conhecimento sobre um determinado domínio. A modelagem de sentenças em linguagem natural ou artificial é relativamente simples por meio de regras fuzzy (JANG et al., 1997). Além disso, as regras fuzzy permitem a representação de conhecimento impreciso, o que facilita a modelagem de problemas do mundo real. As regras, fuzzy ou não, podem ser usadas por sistemas computacionais em seus mecanismos de raciocínio para inferir conclusões a partir de fatos conhecidos.

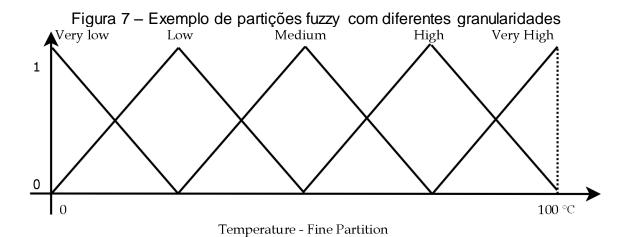
Esta seção descreve os conceitos de variável linguística e regras fuzzy, além dos fundamentos dos processos de inferência que utilizam regras fuzzy.

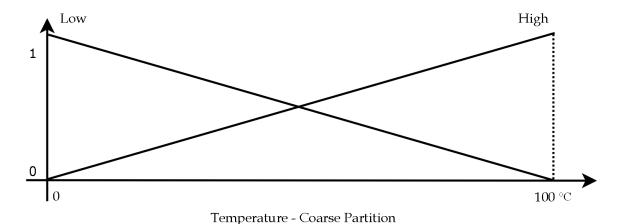
4.2.1 Variáveis lingüísticas

Grande parte da experiência e do conhecimento dos seres humanos pode ser bem representada na forma linguística, mais geral e imprecisa. Esta representação dificulta o uso dessas informações por computadores, que processam informações precisas. As variáveis linguísticas permitem que informações imprecisas e complexas sejam representadas apropriadamente através de termos linguísticos, facilitando o processamento dessas informações por computadores.

Uma variável linguística pode ser definida, de maneira informal, como uma variável cujos valores são palavras ou sentenças em linguagem natural ao invés de números (ZIMMERMANN, 1991). Elas são definidas sobre um conjunto base, que é granularizado em termos (ou valores) linguísticos definidos por conjuntos fuzzy. Este processo de granularização de um domínio em conjuntos fuzzy define uma **partição fuzzy** para a variável em questão. Desta forma, quanto maior for o número de termos linguísticos, maior a granularização da partição (partição fina), e quanto menor o número de termos, menor a granularização da partição (partição (partição grossa). A Figura 7 apresenta uma variável *temperatura* definida por duas partições com granularidades diferentes, a primeira figura representando uma

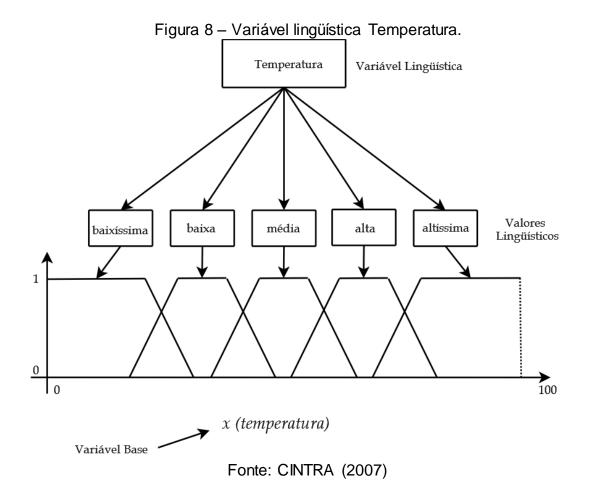
partição fina, com 5 conjuntos fuzzy para representar a variável temperatura, a segunda figura representando a mesma variável com uma partição fina definida por 2 conjuntos fuzzy.





Fonte: CINTRA (2007)

A Figura 8 mostra um exemplo de definição de uma partição para uma variável Temperatura. São usados cinco valores linguísticos para caracterizar a variável Temperatura: baixíssima, baixa, média, alta e altíssima. Estes valores são especificados por conjuntos fuzzy definidos por suas respectivas funções de pertinência. Neste exemplo as funções de pertinência são trapezoidais. Desta forma a variável Temperatura é definida pelo conjunto de termos linguísticos $D_{Temperatura} = \{baixíssima, baixa, média, alta, altíssima\}$.



4.2.2 Sintaxe das regras Fuzzy

As regras são uma das técnicas mais antigas para representação de conhecimento e continuam a ser amplamente utilizadas por serem capazes de expressar claramente diretivas e estratégias, de captar o conhecimento de especialistas humanos e de possuir um formato linguístico de fácil compreensão.

As regras fuzzy são uma maneira fácil para expressar o conhecimento impreciso dando transparência e compreensibilidade aos sistemas fuzzy (KLIR; YUAN, 1995).

Em geral, as regras fuzzy têm a seguinte forma:

$$SE < antecedente > ENTÃO < consequente >$$

e têm a finalidade de estabelecer relações entre as variáveis de seu antecedente, também chamado de condição ou premissa, e as variáveis de seu consequente, também chamado de conclusão ou ação.

A parte antecedente das regras fuzzy é sempre uma proposição fuzzy, ou uma conjunção/disjunção destas. Essas proposições podem ser atômicas ou compostas.

As proposições atômicas especificam um valor linguístico A para uma variável linguística V e tem a forma: V é A.

As proposições compostas podem ser formadas por conjunções ou disjunções de proposições atômicas. Por exemplo, as proposições:

$$X_1 \notin A \mathbf{E} X_2 \notin B$$

е

$$X_1 \notin C OU X_2 \notin D$$

podem representar conhecimento do tipo:

A temperatura está baixa E a umidade está alta;

A água está boa para consumo humano **OU** a inflação está alta.

Uma regra fuzzy assume o formato de uma proposição condicional, que no caso mais simples pode ser expressa como:

SE X é A ENTÃO Y é B

onde X e Y são variáveis lingüísticas sobre os domínios X e Y, respectivamente, e A e B são termos linguísticos, que rotulam conjuntos fuzzy definidos sobre os domínios X e Y, respectivamente.

Em uma regra fuzzy, tanto o antecedente quanto o consequente podem ser formados por proposições compostas, o que dá origem, por exemplo, a regras da forma:

$$\mathbf{SE} \ X_1 \in A_1 \ \mathbf{E} \ X_2 \in A_2 \ \mathbf{E} \ X_3 \in A_3 \ \mathbf{ENT} \tilde{\mathbf{AO}} \ Y_1 \in B_1 \ \mathbf{E} \ Y_2 \in B_2$$

No caso mais geral, tanto o antecedente como o consequente da regra podem ser formados por um número finito de proposições atômicas relacionadas por conjunção ou disjunção. Assim, assumindo que p e q denotam proposições fuzzy,

uma regra fuzzy pode ser da forma **SE** p **ENTÃO** q, sendo que a proposição p pode ter a forma:

$$X_1 \notin A_1 E X_2 \notin A_2 E \dots E X_n \notin A_n, n \ge 1$$

ou

$$X_1 \notin A_1 \cup X_2 \notin A_2 \cup X_n \notin A_n, n \ge 1$$

e a proposição q pode ter a forma:

$$Y_1 \in B_1 \mathbf{E} Y_2 \in B_2 \mathbf{E} \dots \mathbf{E} Y_m \in B_m, m \ge 1$$

ou

$$Y_1 \in B_1 \ \mathbf{OU} \ Y_2 \in B_2 \ \mathbf{OU} \dots \mathbf{OU} \ Y_m \in B_m, m \ge 1$$

onde A_1 , A_2 ,..., A_n e B_1 , B_2 ,..., B_m são conjuntos fuzzy nos universos $X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m$, respectivamente, e X_1, X_2, \ldots, X_n e Y_1, Y_2, \ldots, Y_m são variáveis sobre os domínios $X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m$, respectivamente.

4.2.3 Semântica das regras Fuzzy

A seguir é discutida a semântica das regras fuzzy. Considerando uma regra fuzzy no seu formato mais simples:

entende-se que esta regra descreve uma relação entre as variáveis X e Y, que pode ser descrita por uma relação fuzzy R cuja função de pertinência R(x,y) descreve o grau com que o par $(x,y) \in X \times Y$ é compatível com a relação entre as variáveis X e Y. Diz-se que uma regra fuzzy nessas condições induz uma relação fuzzy R.

Sendo A e B conjuntos fuzzy sobre \mathbf{X} e \mathbf{Y} , respectivamente, a relação R pode ser determinada por:

$$R(x,y) = f(A(x),B(y)), \ \forall (x,y) \in X \times Y$$

na qual f é uma função na forma:

$$f: [0,1]^2 \to [0,1]$$

A escolha da função f é fundamental para o Sistema Fuzzy, pois determina a semântica da regra e diversos aspectos de processamento do sistema. Em geral as relações induzidas por regras fuzzy derivam de três classes principais de funções: conjunção fuzzy, disjunção fuzzy ou implicação fuzzy (PEDRYCZ; GOMIDE, 1998), detalhadas a seguir.

• Conjunção Fuzzy é uma função

$$f_t: [0,1]^2 \rightarrow [0,1]$$

definida por:

$$f_t$$
: $(A(x), B(y)) = A(x)tB(y)$

na qual **t** denota uma *t-norma*.

As definições mais comuns são as conjunções de Mamdani (MAMDANI; ASSILIAN, 1975; MAMDANI, 1977) e Larsen (LARSEN, 1980) que utilizam o mínimo e o produto algébrico, respectivamente.

– Mamdani:

$$f_m(A(x), B(y)) = A(x) \wedge B(y)$$

– Larsen:

$$f_p(A(x), B(y)) = A(x) \cdot B(y)$$

• Disjunção Fuzzy é uma função

$$f_t: [0,1]^2 \to [0,1]$$

definida por:

$$f_n(A(x),B(y)) = A(x)sB(y)$$

na qual **s** denota uma s-norma.

• Implicação Fuzzy é uma função

$$f_t: [0,1]^2 \to [0,1]$$

definida por um conjunto de propriedades e classificada em categorias caracterizadas pelo formalismo da lógica clássica e da lógica intuicionista. A definição formal não é apresentada aqui por não ser fundamental para o objetivo do trabalho. Maiores detalhes podem ser encontrados em (PEDRYCZ; GOMIDE, 1998). Um dos exemplos mais conhecidos de implicação fuzzy é a implicação de Lukasiewicz (LUKASIEWICZ, 1970):

$$f_1(A(x),B(y)) = min[1,1 - A(x) + B(y)]$$

Caso a regra possua múltiplas variáveis no antecedente ou no consequente, a relação R é definida de forma semelhante. Para a regra:

$$\mathbf{SE} \ X_1 \notin A_1 \ \mathbf{E} \ X_2 \notin A_2 \ \mathbf{E} \dots \mathbf{E} \ X_n \notin A_n \ \mathbf{ENT} \tilde{\mathbf{AO}} \ Y_1 \notin B_1 \mathbf{E} \ Y_2 \notin B_2 \ \mathbf{E} \dots \mathbf{E} \ Y_m \notin B_m$$

define-se a relação R assumindo que a proposição composta X_1 é A_1 E X_2 é A_2 E ... E X_n é A_n é induz uma relação n-ária definida por:

$$P_{\alpha}(x_1, x_2, ..., x_n) = A_1(x_1) t A_2(x_2) t ... t A_n(x_n)$$

e a proposição composta Y_1 é B_1 E Y_2 é B_2 E ... E Y_m é B_m induz uma relação m-ária definida por:

$$P_c(y_1, y_2, ..., y_m) = B_1(y_1) \mathbf{t} B_2(y_2) \mathbf{t} ... \mathbf{t} B_m(y_m)$$

A regra induz então uma relação de ordem n + m definida por:

$$R(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) = f(P_\alpha(x_1, x_2, ..., x_n), P_\alpha(y_1, y_2, ..., y_m))$$

Caso o antecedente ou consequente da regra seja formado pelo conectivo ${\bf OU}$ em vez de ${\bf E}$, substituímos a t-norma por uma s-norma no cálculo das relações induzidas P_a ou P_c .

4.2.4 Inferência com regras Fuzzy

A computação com regras fuzzy envolve a aplicação de um método de raciocínio que está baseado em regras de inferência. A regra da inferência composicional é a regra de inferência básica para os processos de raciocínio nos sistemas fuzzy. A regra da inferência composicional, também chamada de regra da composição, será apresentada para o caso simples, isto é, quando a regra tem apenas uma variável no antecedente e uma no consequente.

O esquema de inferência de interesse nesse contexto é aplicado a partir de uma regra, que estabelece um conhecimento genérico sobre algum domínio e de um fato, que estipula uma informação específica sobre a situação da qual se deseja inferir alguma conclusão.

Seja A um conjunto fuzzy sobre X, e B um conjunto fuzzy sobre Y. Conforme o que foi descrito anteriormente sobre a semântica das regras, a regra:

SE X é A ENTÃO Y é B

induz uma relação fuzzy R(x,y) sobre $X \times Y$. Para usar essa regra em uma inferência, é necessário conhecer algum fato que estabeleça um valor específico para a variável do antecedente X.

Dado o fato $X \notin A'$, pode-se inferir a conclusão $Y \notin B'$, sendo que $B' \notin$ definido por (PEDRYCZ; GOMIDE, 1998):

$$B'(y) = \sup_{x \in X} [A'(x) \mathbf{t} R(x, y)] \forall y \in \mathbf{Y}$$

na qual *sup* denota o operador de supremo e **t** é uma *t-norma*.

A aplicação da regra da inferência composicional pode ser facilmente estendida para regras com múltiplas variáveis e também para situações em que o

conhecimento do problema é descrito por um conjunto de regras. O processo geral de computação usando regras fuzzy é baseado na regra de inferência composicional.

Alternativamente a este, pode ser aplicado o chamado método de inferência escalonada, que envolve um número pequeno de cálculos numéricos com relação ao método geral. Esses dois processos gerais de computação com regras estão descritos em (PEDRYCZ; GOMIDE, 1998). O processo geral de inferência e a inferência escalonada não serão apresentados aqui, uma vez que não são utilizados no sistema estudado. A discussão apresentada focaliza os métodos de raciocínio específicos para problemas de classificação, que é o foco principal deste trabalho.

4.3 SISTEMAS FUZZY BASEADOS EM REGRAS

Um Sistema Fuzzy é qualquer sistema que incorpora algum mecanismo derivado da lógica fuzzy e que tenha pelo menos uma variável que possa assumir valores linguísticos definidos por conjuntos fuzzy (KLIR; YUAN, 1995).

O tipo particular de Sistema Fuzzy de interesse para este trabalho são os Sistemas Fuzzy Baseados em Regras, ou apenas Sistema de Regras Fuzzy. Os SFBR têm sido aplicados com sucesso em diversas áreas, como controle automático, classificação e reconhecimento de padrões (ICHIBA et al., 2006; MAGDALENA; MONASTERIO, 1997; CHUNG et al., 2000; CHI et al., 1996; DUMITRESCU et al., 2000). Os SFBR possuem dois componentes principais: a Base de Conhecimento (BC) e o Mecanismo de Inferência (MI).

A BC é composta pela Base de Dados Fuzzy, que contém as definições dos conjuntos fuzzy relacionados aos termos linguísticos usados nas regras fuzzy, e pela Base de Regras Fuzzy, que armazena o conjunto de regras fuzzy que modelam um determinado problema.

O Mecanismo de Inferência é responsável pelo processamento das regras, o qual é realizado por um método de raciocínio. Este consiste da aplicação de um procedimento de inferência para derivar conclusões a partir das regras e de fatos conhecidos.

Com a definição dos diversos parâmetros envolvidos em SFBR obtém-se o que é chamado de *modelo de sistema fuzzy*, sendo que alguns desses modelos recebem destaque na literatura devido a seu sucesso em aplicações práticas. Nesse

sentido, pode-se citar dois modelos principais de sistemas fuzzy: os de Mamdani (MAMDANI; ASSILIAN, 1975; MAMDANI, 1977) que possuem na parte consequente das regras fuzzy uma proposição fuzzy e os de Takagi-Sugeno (TAKAGI; SUGERO, 1983, 1985) que possuem na parte consequente uma função aplicada sobre os valores de entrada.

4.3.1 Sistemas Fuzzy de classificação

Classificação é uma importante tarefa encontrada nas áreas de reconhecimento de padrões, tomada de decisão, mineração de dados e modelagem (JAIN; MARTIN, 1998; SANCHEZ et al., 1997). Dado um conjunto de objetos $E = \{e_1, e_2, \ldots, e_N\}$, também chamados de padrões, o objetivo da classificação é atribuir uma classe C_j de um conjunto de classes $C = \{C_1, C_2, \ldots, C_m\}$ a um objeto e_p , $e_p = (a_{p1}, a_{p2}, \ldots, a_{pk})$ o qual é descrito por k atributos.

Muitos métodos têm sido utilizados para a tarefa de classificação de padrões. Entre eles destacam-se os métodos estatísticos (DUDA; HART, 1973), as redes neurais (BISHOP, 1995) e os sistemas fuzzy (KECMAN, 2001).

Um Sistema de Classificação Fuzzy Baseado em Regras (SCFBR) é um SFBR cujas regras foram projetadas para resolver um problema de classificação.

Uma típica regra fuzzy de classificação pode ser expressa por:

$$R_k : \mathbf{SE} X_1 \in A_{1l_1} \mathbf{E} X_2 \in A_{2l_2} \mathbf{E} \dots \mathbf{E} X_n \in A_{nl_n} \mathbf{ENT} \tilde{\mathbf{AO}} \mathbf{Classe} = C_j$$

na qual R_k é o identificador da regra, X_1, \ldots, X_n os atributos do padrão considerado no problema (representados aqui por variáveis linguísticas), $A_{1l_1}, \ldots, A_{nl_n}$ os valores linguísticos usados para representar os valores de tais atributos e C_j a classe, fuzzy ou não, a que pertence o padrão.

4.3.2 Métodos de inferência

O **Mecanismo de Inferência** aplica o conjunto de regras fuzzy ao padrão a ser classificado, determinando a classe a que ele pertence. A maioria dos sistemas fuzzy de classificação utiliza o *Método de Raciocínio Fuzzy Clássico*, também

chamado de *Método da Regra Vencedora* (CHI et al., 1996; GONZALEZ; PIREZ, 1999), que classifica um padrão usando a regra que individualmente possuir maior grau de compatibilidade com esse padrão. Já o *Método de Raciocínio Fuzzy Geral* agrega os graus de compatibilidade das regras da BR com o padrão que tenha a mesma classe consequente. Desta forma, um padrão é classificado de acordo com a classe com maior grau de compatibilidade com relação a ele, considerando todas as regras da mesma classe. A seguir são descritos os métodos de raciocínio fuzzy clássico e geral.

4.3.2.1 Método de Raciocínio Fuzzy Clássico

Seja $e_p=\left(a_{p_1},a_{p_2},\ldots,a_{p_n}\right)$ um padrão a ser classificado e $\{R_1,R_2,\ldots,R_s\}$ o conjunto de s regras de um sistema de classificação fuzzy, cada uma com n antecedentes. Seja $A_{1l_1}\left(a_{p_i}\right)$, $i=1,\ldots,n$, o grau de pertinência do valor do atributo a_{p_i} no i-ésimo conjunto fuzzy da regra fuzzy R_k . O método de raciocínio fuzzy clássico aplica os seguintes passos para classificar o padrão e_n :

1. Calcular o grau de compatibilidade entre o padrão e_p e cada regra R_k , para $k=1,\ldots,s$.

$$Compat(R_k, e_p) = t(A_{1l_1}(a_{p_1}), A_{2l_2}(a_{p_2}), \dots, A_{nl_n}(a_{p_n}))$$

na qual *t* denota uma *t-norma*.

2. Encontrar a regra $R_{\rm k_{max}}$ com o maior grau de compatibilidade com o padrão:

$$max\{Compat(R_k, e_p)\}, k = 1, 2, ..., s$$

3. Atribuir a classe C_j ao padrão e_p , na qual C_j é a classe predita pela regra $R_{\mathbf{k}_{\max}}$ encontrada no passo anterior.

A Figura 9 mostra graficamente o Método de Raciocínio Fuzzy Clássico. A compatibilidade do padrão de entrada é calculada com relação a todas as s regras do conjunto de regras, e a ele é atribuída a classe C_j da regra R_k , que possui o maior grau de compatibilidade com o padrão.

Figura 9 – Raciocínio fuzzy clássico

R 1

R 2

R 3

Class c_j

R _k

R _s

4.3.2.2 Método de Raciocínio Fuzzy Geral

Seja $e_p=\left(a_{p_1},a_{p_2},\ldots,a_{p_n}\right)$ um padrão a ser classificado e $\{R_1,R_2,\ldots,R_s\}$ o conjunto de s regras de um sistema de classificação, cada uma com n antecedentes. Seja $A_{1l_1}\left(a_{p_i}\right)$, $i=1,\ldots,n$, o grau de compatibilidade do valor do atributo a_{p_1} no i-ésimo conjunto fuzzy da regra fuzzy R_k . O método de raciocínio fuzzy geral aplica os seguintes passos para classificar o padrão e_p :

Fonte: CINTRA (2007)

1. Calcular o grau de compatibilidade entre o padrão e_p e cada regra R_k , para $k=1,\ldots,s$.

$$Compat(R_k, e_p) = t(A_{1l_1}(a_{p_1}), A_{2l_2}(a_{p_2}), \dots, A_{nl_n}(a_{p_n}))$$

na qual t denota uma t-norma.

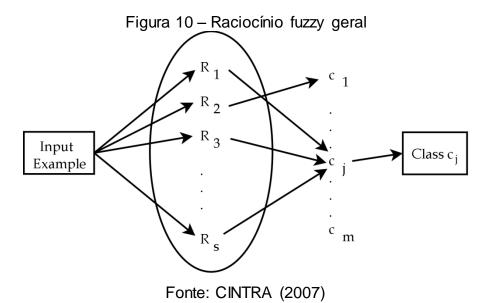
2. Para cada classe C, calcular o valor de $Classe_C$, ou seja, o grau de classificação do padrão na classe, agregando os graus de associação do passo anterior de todas as regras cuja classe predita é C:

$$Classe_{C} = f \{Compat(R_{k}, e_{p}) | C \in a classe de R_{k}\}$$

sendo f um operador de agregação tal que $min \leq f \geq max$.

3. A classe C_j será atribuída ao padrão e_p , sendo C_j a classe de $Classe_C$ que possui a maior soma de graus de pertinência.

A Figura 10 mostra graficamente o Método de Raciocínio Fuzzy Geral. A compatibilidade do padrão de entrada é calculada com relação a cada uma das n regras existentes e os graus de compatibilidade das regras que levam a mesma classe são agregados. A classe C_j que possui o maior valor resultante da agregação é atribuída ao padrão.



No próximo capítulo é apresentada a revisão bibliográfica deste trabalho.

5 REVISÃO BIBLIOGRÁFICA

Várias iniciativas na área de jogos sérios para ensino de engenharia de software têm sido realizadas. Muitos desses jogos são do tipo tradicional, envolvendo tabuleiros ou cartões. Há também jogos desenvolvidos para computadores.

Entretanto, existe uma carência especificamente de jogos para a aprendizagem da gestão de risco em projetos de software. A seguir, são apresentados alguns dos jogos mais relevantes relacionados à proposta desse trabalho, em ambas as categorias, jogos tradicionais e jogos para computadores.

Neste capítulo é apresentada algumas das propostas de jogos educativos sobre o ensino de gerência de projetos de software mais relevantes para esse trabalho. O objetivo desse estudo é obter informações relevantes para a concepção de um jogo educativo de computador para apoiar o ensino de gerenciamento de riscos em projetos de software.

A pesquisa dos trabalhos é realizada com base em publicações de revistas, bibliotecas digitais, teses, dissertações e monografias em Computação.

5.1 JOGOS TRADICIONAIS E DE TABULEIRO PARA O ENSINO DE ENGENHARIA DE SOFTWARE

Existe uma quantidade e diversidade grande de jogos tradicionais para ensino da computação, incluindo jogos relacionados ao tema de Engenharia de software e mais especificamente a Gestão de Riscos.

O Lego SCRUM City (AGILE, 2013) é um jogo de tabuleiro para o ensino de desenvolvimento de software no qual os alunos jogam em grupos e recebem artefatos usados em sistemas geridos pelas metodologias ágeis. O jogo se baseia na requisição e construção de objetos com peças do jogo LEGO. Todo o projeto é baseado no processo de desenvolvimento SCRUM (SCHWABER; BEEDLE, 2001). O Scrum é um processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de software.

Outro jogo tradicional para ensino de projeto de softwares é o Project Detective (von WANGENHEIM et al., 2011), no qual os jogadores simulam que são contratados como consultores de uma empresa para descobrir o que ocorreu de

errado em um cenário hipotético de um projeto de software. No início do jogo, os jogadores recebem vários documentos para serem analisados, e a partir deles, identificarem quais problemas ocorreram.

O SCRUMIA (SAVI, 2011) é um jogo cujo objetivo é o aprendizado do gerenciamento de projetos usando a metodologia SCRUM. O processo é desenvolvido por meio de uma competição usando um projeto hipotético de construção de objetos genéricos (chapéus, barcos e aviões de papel). Em nível cognitivo, o objetivo de aprendizagem da atividade é reforçar os conceitos e ensinar competências necessárias à aplicação de gerenciamento de projetos utilizando a metodologia do processo.

O Jogo de Gerenciamento de Riscos (TARAN, 2007) é multi-usuário e tem como finalidade ensinar conceitos de Gerência de Riscos em projetos de softwares através da simulação. Nele, cada jogador assume o papel de gerente de projeto e competem uns contra os outros, desenvolvendo um produto e vendendo no mercado. Vence quem tem mais dinheiro no fim do jogo. Apresenta cinco fases: planejamento, requisitos, arquitetura e design, implementação e testes; cinco conjuntos de cartas principais: cartas de projeto, surpresa, *oops*, de risco e de mitigação. Os jogadores iniciam com um conjunto de recursos (pessoas e dinheiro) e um conjunto de 5 cartas de risco. Em cada rodada podem fazer uma de duas coisas: executar um passo do projeto passo ou mitigar um risco. Independentemente do que fazem tem um custo em dinheiro e pessoas associado.

5.2 JOGOS DE COMPUTADOR PARA O ENSINO DE ENGENHARIA DE SOFTWARE

É possível encontrar na literatura algumas propostas de jogos de computador relacionadas ao ensino de engenharia de software. O jogo TIM: *The Incredible Manager* (DANTAS et al., 2004), para um único jogador, envolve uma simulação com foco em planejamento e controle do orçamento, cronograma e qualidade na gestão da qualidade de software. O jogador atua como gerente de projetos que possui tarefas que devem ser atribuídas a cada desenvolvedor, além de determinar o tempo limite para a conclusão da tarefa e o número de inspeções, entre outros, para finalmente apresentar esta especificação para uma possível aceitação.

Em (AGARWAL; UMPHRESS, 2010) um modelo de simulação de desenvolvimento de software que utiliza o paradigma de jogos é apresentado. Nesse trabalho, os agentes são usados para representar os desenvolvedores e cada agente tem um padrão de comportamento diferente, montado a partir de estatísticas do sistema. O usuário monta a equipe e especifica os dados de desempenho e produtividade do desenvolvedor. O desenvolvimento do jogo se dá seguindo um processo de programação extrema (XP - eXtreme Programming) (SHAHZAD, 2009).

Segundo Beck (2004), seu criador, a *Extreme Programming* é uma metodologia ágil para equipes médias e pequenas, onde os requisitos para o desenvolvimento de software são vagos e em constante mudança.

Um projeto semelhante é o Kallango (CAMPOS et al., 2011), que usa um jogo educacional seguindo o processo SCRUM (SCHWABER; BEEDLE, 2001). Onde o jogo simula que o usuário-jogador é contratado por uma empresa como gerente de projetos. Para desempenhar seu papel, o jogador conta com a participação de personagens virtuais, colaboradores da empresa. O jogador deve, então, aplicar as práticas do Scrum para manter o projeto em andamento e alcançar metas pré-estabelecidas, bem como avaliar e resolver problemas que ocorrerão no desenrolar do projeto. Assim, à medida que o jogador vai resolvendo os problemas, os conceitos relacionados à metodologia Scrum vão sendo experimentados, e as consequências de suas práticas sendo avaliadas.

O Jogo da Equipe de Teste de Software (JETS) (SILVA et al., 2011), que é multi-usuário, visa apoiar o processo de ensino-aprendizagem relacionado ao tema de teste de softwares em cursos de graduação na área de computação. O JETS visa simular as interações de uma equipe de teste de uma empresa de desenvolvimento de software.

Já o jogo SimSE (NAVARRO; van der HOEK, 2004), desenvolvido em Java, foca no gerenciamento de projeto de software, englobando as fases de concepção, análise, projeto, construção e testes. O jogador do SimSE assume o papel de um gerente de projetos. Executa tarefas e toma decisões desde o planejamento à entrega do software. O fim do jogo é dado com o release do software completo ou não, dentro do cronograma previsto e com o menor numero de erros. Uma pontuação baseada em seu desempenho é dada quando entrega o software no prazo.

O Planager (KIELING; ROSA, 2006) é jogado por apenas um jogador e tem o foco tanto no planejamento quanto no gerenciamento do escopo e do tempo. O usuário joga em cenários já cadastrados passando por diversas fases e sendo avaliado ao final de cada uma. O Planager oferece ranking de pontuações no final do jogo para demonstrar o desempenho do jogador.

Por fim, o PMK Learning Environmemnt (TORREAO, 2005) é um ambiente inteligente de aprendizagem em gerenciamento de projetos e preparação para certificação PMP. Nele o jogador é cadastrado de acordo com sua personalidade e grau de conhecimento em gerencia de projetos. Os estudos são individuais, em etapas e auxiliados pela comunidade on-line através de fórum de discussão, *chat* e do agente inteligente *VICTOR*.

5.3 JOGOS DE COMPUTADOR PARA GESTÃO DE RISCOS

Como dito anteriormente, apesar de haver na literatura várias propostas de jogos sérios para aprendizado focados em diversas áreas da computação, especificamente para o ensino de gestão de risco há uma carência desse tipo de jogo.

O RSKManager (PALUDO, 2013) é um jogo de computador para um usuário que aborda a identificação, a análise, o planejamento de respostas, o monitoramento e a execução de ações de tratamento de riscos em projetos de software. No jogo, os riscos são considerados como positivos ou negativos em relação aos objetivos do projeto de software abordado. Os problemas que o jogador deve resolver, o contexto ou história do jogo, são apresentados por estudos de caso de projetos de software configurados pelo professor da disciplina a partir de experiências, contextos relevantes da realidade, nível de complexidade que deseja abordar, customizando restrições de prazo, custo, contexto do projeto, riscos, análise e respostas adequadas aos riscos. Esses parâmetros são utilizados para o feedback dado ao aluno de seu desempenho no jogo. Vence o jogo, o jogador que finalizar o projeto dentro dos critérios de aceitação planejados no estudo de caso, simulando a entrega do projeto para o cliente.

Outra proposta é o eRiskGame (GALVAO, 2012), um jogo sério multiusuário voltado para o aprendizado de gerência de projetos. O foco do SuperRisk é o gerenciamento do controle de gastos, para o cumprimento das metas

e prazos estabelecidos, além de acompanhar a produtividade da equipe de trabalho. Porém essa proposta tem deficiências relacionadas à simulação da realidade, carência de processos automáticos e interação entre jogadores.

A proposta apresentada nesse trabalho de mestrado, *SuperRisk*, têm pontos de avanço em relação aos trabalhos similares disponíveis na literatura. O *SuperRisk* é detalhado no próximo capítulo.

6 SUPERRISK

Nesse capítulo é descrita a proposta deste trabalho para o ensino e treinamento em gerência de riscos no âmbito de projetos por meio de um jogo sério, SuperRisk.

O SuperRisk é um jogo sério para o ensino de gestão de projetos de software baseado em navegadores para a internet. O SuperRisk engloba várias tarefas que um gerente de projeto deve executar em uma organização. Com foco na gestão de riscos, mais especificamente nas tarefas de planejamento, controle e monitoramento, incluindo o orçamento e cronograma. O SuperRisk usa Aprendizagem Baseada em Projetos (ABP) (AYAS; ZENIUK, 2003) para possibilitar ao jogador a aquisição de conhecimento em gerenciamento de projetos, especialmente no controle dos riscos envolvidos neste processo.

O *SuperRisk* permite que alunos interajam em um ambiente que simula os principais aspectos importantes da gestão de risco de forma atraente e agradável.

6.1 CENÁRIO E ENREDO DA SIMULAÇÃO

Toda a simulação apresentada no jogo SuperRisk é baseada no cenário de uma empresa de software que concorre com outras companhias pelo desenvolvimento de projetos. Cada participante é responsável pelo gerenciamento de uma equipe de desenvolvimento de software de uma empresa. Durante o decorrer do tempo, essa empresa pode passar por mudanças, criando riscos que devem ser tratados.

Os jogadores recebem projetos de software que devem ser planejados e desenvolvidos. Esses projetos são divididos em fases distintas, com prazos, orçamentos e metas previamente estabelecidas pelo administrador, as quais permitem a avaliação do andamento dos mesmos. Assim, os jogadores, como gerentes de projetos, devem acompanhar seus projetos e tomar decisões. Para isso, são fornecidas estatísticas de acompanhamento dos projetos e equipes, em tabelas e gráficos.

Com o objetivo de simular a experiência de gestão de risco da forma mais real possível, cada jogador participa do gerenciamento das seguintes tarefas:

- Contratação de profissionais para integrarem as equipes de trabalho com base na análise de características objetivas de cada profissional disponível. O SuperRisk provê uma lista de profissionais disponíveis no mercado do jogo. Esses profissionais são disputados pelos jogadores;
- Fazer proposta de salários aos funcionários já contratados, com o objetivo;
- Demissão de profissionais com base na análise de desempenho dos mesmos;
- Aquisição produtos para qualificar os funcionários de sua equipe de trabalho;
- 5. Controle de gastos do orçamento dos projetos;
- Definição de estratégias para cumprir metas e prazos do desenvolvimento do projeto;
- 7. Previsão de riscos relacionados aos projetos;
- 8. Comunicação entre jogadores para troca de experiências.

O *SuperRisk* apresenta um aspecto lúdico de gerência de riscos, cujo objetivo é cativar o jogador de modo a lhe apresentar conceitos inconscientemente. A fixação de tais conceitos acontece quando os jogadores os aplicam para transpor os desafios impostos durante o jogo.

Note que assim como no mundo real, os jogadores disputam no mercado simulado do *SuperRisk* a contratação de funcionários por meio de uma negociação de salários. Assim, mesmo os funcionários contratados em outras empresas podem receber propostas e mudarem de equipe.

Os funcionários das equipes de trabalho no *SuperRisk* estão sujeitos à condições similares do mundo real, como situações de falta de motivação no trabalho, afastamentos por doença, solicitações de demissão, ofertas de outras empresas e até mesmo baixa de produtividade. Essas condições podem ocorrer de acordo com probabilidades definidas em relação às características da empresa. Sendo assim, para aumentar os atributos, como a motivação e produtividade de sua equipe, o jogador deve fazer aquisições de produtos que incluem:

- 1. Palestras motivacionais;
- 2. Cursos de aprimoramento;
- 3. Livros;

- 4. Computadores;
- 5. Internet melhor;
- 6. Máquina de café;
- 7. Impressora;
- 8. Contratação de hora extra.

Cada aquisição feita no jogo tem influência direta nos atributos dos funcionários, isto é, o jogador pode aumentar a produtividade de sua equipe em um curto espaço de tempo por meio de horas extras. Porém, as horas extras ocasionam uma queda de motivação dos funcionários. Dessa forma, o jogador deve dispor de uma estratégia para contratação desse serviço apenas em momentos de necessidade.

Durante o jogo as ações do jogador geram uma pontuação. E essa pontuação tem como objetivo, além do reforço positivo, provocar uma competição entre os jogadores. Essa competição, por meio da comparação das ações entre os diversos jogadores, torna a simulação do *SuperRisk* divertida e motiva a participação.

6.2 VISÃO GERAL DO SUPERRISK

Como explicado anteriormente, o foco da proposta apresentada neste trabalho é o ensino de Gerência de Riscos, visto que uma das maiores dificuldades de gerentes de projetos é prever os problemas que podem surgir durante o desenvolvimento de um projeto e definir planos para minimizar seus efeitos.

A simulação de um ambiente real de gerência de projetos é bastante complexa e de difícil representação, tendo em vista envolver diversas tarefas relacionadas aos riscos do projeto, produto e negócio. Para auxiliar na definição deste ambiente, foram usados agentes inteligentes (RUSSELL; NORVIG, 2004) para simular riscos em etapas e condições diferentes em cada projeto.

O SuperRisk foi desenvolvido em módulos de funcionalidades, cada um deles com atribuições específicas. Os detalhes mais técnicos do \erg incluindo telas dos módulos são descritos no Anexo A.

No SuperRisk seus módulos interagem entre si para simular um ambiente real de gerência de projetos de software. A Figura 11 apresenta o esquema geral do

SuperRisk por meio do Diagrama de Atividades da UML (*Unified Modeling Language*) (BOOCH et al., 1999).

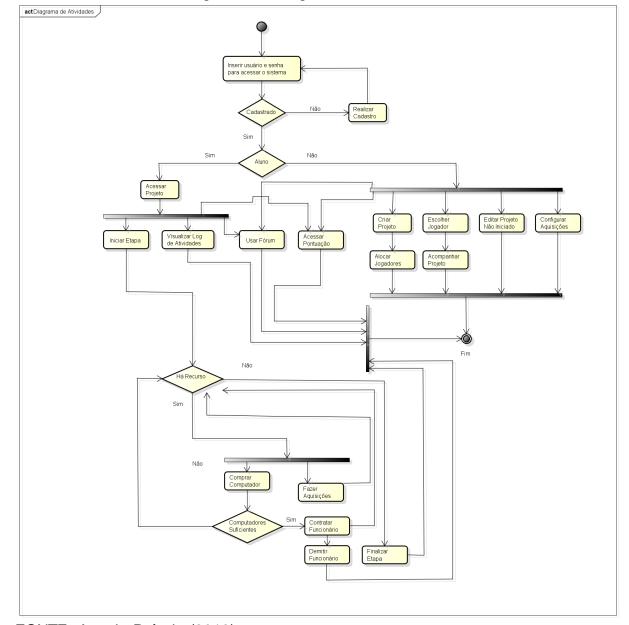


Figura 11 – Diagrama de atividades.

FONTE: Autoria Própria (2013).

6.2.1 Funcionamento do SUPERRISK

O *SuperRisk* é iniciado solicitando a identificação do usuário, caso o aluno ainda não possua é possível fazer um breve cadastro. Existem dois tipos de usuários no *SuperRisk*, são eles:

1. Administrador: Caracterizado pela figura do professor da disciplina.

2. **Jogador**: Caracterizado pela figura do aluno.

Cada um desses tipos de usuários possui atribuições específicas que devem ser seguidas no jogo. Essas atribuições são descritas a seguir.

6.2.1.1. Administrador

O Administrador do *SuperRisk* tem a função de cadastrar os projetos e as etapas desses projetos com as especificações de prazo, meta, orçamento, data e hora de início para cada projeto. Além disso o administrador deve alocar quais os jogadores participarão daquele determinado projeto.

Acompanhar o desempenho de cada um dos jogadores também é uma atividade do administrador do *SuperRisk*, para realizar essa tarefa todos os projetos e jogadores serão listados para a escolha do administrador.

6.2.1.2. Jogador

O jogador do *SuperRisk*, acessa o jogo e verifica quais projetos estão disponíveis para ele. Após essa verificação é possível acessar o projeto e iniciar as etapas. Cada etapa tem um recurso pré-estabelecido, recurso esse que deve ser gasto com aquisições de equipamentos e funcionários para a sua equipe de desenvolvimento. Um detalhe importante é que para contratar funcionários para a equipe de trabalho, o jogador deve adquirir a mesma quantidade de computadores.

Com o decorrer do jogo, o jogador pode contratar e demitir funcionários da sua equipe de trabalho. Esses funcionários possuem atributos que são mutáveis no decorrer do jogo, podendo até ficar afastado por licença médica.

O jogo é finalizado quando o jogador atinge as metas estabelecidas pelo administrador no ato do cadastro do projeto e suas etapas ou quando o projeto é finalizado pelo administrador.

Maiores detalhamentos sobre os módulos e imagens do *SuperRisk* são descritos no Anexo A dessa dissertação.

6.3 DESCRIÇÃO DOS AGENTES

Devido à diversidade dos problemas encontrados em uma situação de gerência de risco, cinco agentes foram implementados para simular ambiente o \erg. Cada um dos agentes é responsável por monitorar um aspecto específico do jogo e assim, cada agente atua em relação a um risco específico. Cada um desses agentes recebeu um nome para facilitar a iteração com os mesmos. Os agentes e suas respectivas atividades são:

- Agente Dias: Criação e contagem de dias no projeto, verificação da produção da equipe de trabalho e cálculo da pontuação dos jogadores;
- Agente Nascimento: Controle da criação de novos funcionários com características distintas e inserção dos funcionários no mercado para contratação, de acordo com a necessidade apresentada no cenário dos projetos dos jogadores e simular a negociação salarial entre o jogador e o funcionário;
- Agente das Chagas: Afastamento de funcionários por motivo de doença;
- Agente Sara: Diminuição da quantidade de dias de afastamento dos funcionários ausentes por motivo de doença;
- 5. Agente Costa: Controle dos seguintes fatores comportamentais dos funcionários: concentração, produtividade, motivação, vigor e trabalho em equipe além de gerar pedidos de demissão dos funcionários durante o projeto.

Para simular todos estes aspectos do mercado de produção de software, algumas técnicas diferentes de IA foram utilizadas com a finalidade de prover inteligência às decisões dos agentes. O *SuperRisk* usa um Sistema Fuzzy baseado em regras para inserir conhecimento específico na forma de regras.

A seguir, são descritas todas as lógicas de cada um dos agentes inteligentes.

6.3.1 Agente Dias

O Agente Dias é o agente principal do jogo e tem como função iniciar os projetos e contabilizar os dias decorridos durante o passar do tempo. Note que a contagem de dias no jogo é diferente da contagem de dias no mundo real. Porém essa definição pode ser alterada pelo administrador de acordo com sua disponibilidade de tempo.

O ciclo do agente Dias é feito de tempos em tempos, de acordo com a configuração realizada pelo administrador. Seu funcionamento é iniciado pela comparação da data em que o projeto deve iniciar com a hora do servidor no qual o jogo está armazenado. Se após a comparação for verificado que a data foi alcançada, é iniciado o projeto, alterando o *status* de "não inicializado" para "em andamento". Juntamente com a mudança do status, um dia é criado no projeto.

Independentemente de quantos projetos tenham sido cadastrados, a contagem dos dias é sempre igual para todos, ou seja, os jogadores terão os dias alterados no mesmo momento.

No momento da próxima mudança de dia, este agente novamente se comunica com o sistema e cria um novo dia. Esse processo se repete até que a quantidade máxima de dias do projeto seja atingida.

O agente Dias verifica o dia do projeto em que o jogador iniciou uma determinada etapa. Esse procedimento serve para contabilizar a quantidade de dias já usados pelo jogador. Caso essa quantidade de dias seja maior que a definida no cadastro, o status da etapa é alterado para "atrasado" e o tempo para realizar a etapa seguinte tem decremento da mesma quantidade dos dias atrasados. Isso ocorre até que o administrador do jogo finalize a etapa. Dessa forma é possível simular o cancelamento do projeto por parte do cliente. Quando isso acontece, o status do projeto é alterado para "finalizado" e a pontuação do jogador é zerada, ou seja, o jogador não atingiu as metas propostas.

O agente Dias também verifica, a cada mudança de dia, o somatório da produtividade dos funcionários contratados em cada projeto e compara com a produção total de cada etapa com a finalidade de descobrir se o jogador atingiu a meta definida. Ele também faz a verificação do que foi gasto com salários e aquisições naquele dia e desconta do valor do orçamento da etapa.

Na ocorrência de algum jogador não conseguir alcançar o objetivo proposto pelo administrador do jogo, tanto por falta de recurso financeiro ou por não atingir a produção, o agente Dias baixa a pontuação do jogador no decorrer do projeto.

O Agente Dias também é responsável por um processo muito importante no jogo, o cálculo da pontuação dos jogadores que é responsável por definir a pontuação dos mesmos no jogo.

O processo de cálculo da pontuação dos jogadores é detalhado a seção 6.4.1.

6.3.2 Agente Nascimento

O agente Nascimento tem um ciclo independente do Agente Dias. Ele tem a função de criar e verificar a necessidade de criar novos funcionários no mercado de trabalho do jogo.

O Agente Nascimento também analisa se a quantidade de funcionários disponíveis no mercado é suficiente para o número de jogadores em cada projeto. Para isso, o Agente Nascimento verifica o limite definido pela dificuldade da etapa cadastrado pelo administrador. Essa definição dos limites é feita da seguinte forma:

- 1. Nível Fácil: Quando esse nível de dificuldade é escolhido na etapa, uma quantidade suficiente de funcionários é inserida no mercado de trabalho, não permitindo que o mercado entre em escassez de funcionários. O objetivo deste nível é abordar a capacidade do jogador em realizar boas escolhas de funcionários;
- 2. Nível Médio: Quando esse nível de dificuldade é escolhido na etapa, o mercado de trabalho já apresenta uma quantidade de funcionários reduzida em comparação ao nível mais fácil. Essa opção analisa a capacidade do jogador em modelar os funcionários de acordo com suas necessidades do projeto, fazendo aquisições para aumentar seus atributos;
- 3. Nível Difícil: Quando esse nível de dificuldade é escolhido na etapa, o mercado de trabalho apresenta uma boa escassez de funcionários. Essa opção promove uma competição alta entre os jogadores para contratar funcionários.

Cada funcionário possui atributos com pontuações específicas, onde essas pontuações influenciam vários fatores no decorrer do jogo. Os atributos são:

Nome Identificador de cada funcionário é composto pela

união de um nome e um sobrenome pré-

cadastrado;

Custo Diário Valor, em dinheiro, referente ao valor diário de

cada funcionário:

Produtividade Valor da produtividade diária definido em artefatos

de software ou linhas de código;

Trabalho em Equipe Valor referente à característica de trabalho em

equipe;

Motivação Valor definido em um intervalo de 0 a 100 que

indica o estado de motivação de cada funcionário;

Vigor Valor definido em um intervalo de 0 a 100 que

indica o estado de vigor de cada funcionário;

Liderança Valor referente à capacidade de liderança de cada

funcionário;

Concentração Valor referente à concentração do funcionário;

Experiência Valor referente à experiência do funcionário;

Dias de Afastamento Quantidade de dias que o funcionário deve ficar

afastado por motivo de licença médica. Esse parâmetro sempre é inicializado com zero. Caso o

Agente Das Chagas, atue, esse parâmetro é

alterado.

O Agente Nascimento também simula a negociação salarial entre o jogador e o funcionário, seu funcionamento é descrito na seção 6.4.2.

6.3.3 Agente Das Chagas

Esse agente é responsável por simular o afastamento dos funcionários contratados pelos jogadores. Para simular o ambiente real, um Sistema Fuzzy Baseado em Regras é usado e acionado de acordo com um fator de probabilidade pré-definido. Esse fator representa a probabilidade de um contratempo (risco)

ocorrer. Assim um número aleatório é obtido e comparado com este atributo para assim determinar se o agente irá ou não atuar. Dessa forma, o elemento sorte, ou acaso, é inserido no ambiente de gerencia.

Na seção 6.4.3 todo o processo do sistema fuzzy é detalhado.

6.3.4 Agente Sara

Com a quantidade de dias de afastamento dos funcionários definida pelo Agente Das Chagas, o funcionário precisa retornar à equipe de trabalho. Esse processo é realizado pelo Agente Sara que tem o mesmo ciclo de funcionamento do agente Dias, sendo iniciado a cada mudança de dia no jogo.

A atividade do Agente Sara é selecionar todos os funcionários que estão afastados por licença médica e diminuir um dia do atributo "Dias de Afastamento" do funcionário a cada mudança de dia. Essa ação é executada para cada funcionário afastado até que o referido atributo seja igual a zero, isto é, até que o funcionário conclua todos os dias de afastamento, sendo considerado saudável. Ao retornar à equipe de trabalho, o funcionário tem o valor de vigor e motivação restaurados para o valor máximo.

6.3.5 Agente Costa

A incerteza existente em um ambiente de trabalho relacionada ao comportamento da equipe de trabalho também é simulada no *SuperRisk*. O responsável por esta tarefa é o Agente Costa, ele analisa cada equipe de trabalho e modifica os atributos dos funcionários, fazendo com que exista essa incerteza na simulação.

O Agente Costa é iniciado a cada começo de dia e a partir disso ele seleciona todas as equipes de trabalho na qual os jogadores não estão investindo no bem estar da equipe. Para isso o Agente Costa seleciona a equipe com menor média do atributo "trabalho em equipe".

Após definir a equipe de trabalho, o Agente Costa altera os atributos "Motivação", "Liderança", "Concentração" e "Experiência" de cada um dos funcionários participantes dessa equipe.

Para essa mesma equipe que foi afetada, aleatoriamente um funcionário também tanto pode pedir aumento quanto pode pedir demissão de sua equipe de trabalho.

6.4 DETALHAMENTO DOS SISTEMAS INTELIGENTES

Os processos descritos a seguir contém conhecimento do domínio por meio de regras.

6.4.1 Cálculo da pontuação do Jogador

Para uma melhor análise dos resultados, foi realizada uma abordagem quantitativa para estabelecer a pontuação dos jogadores. Porém esse cálculo não é direto, ou seja, o seu valor depende de alguns fatores, e o seu cálculo é realizado de acordo com o desempenho do jogador ao controlar o orçamento, tempo e produção do projeto, prevendo e evitando riscos. Esses quesitos são descritos a seguir. Cada item tem peso igual no cálculo dos pontos de cada jogador.

Pontuação pelo controle do Orçamento: As ações do jogador, relativas ao gasto de recursos financeiros de cada etapa, são pontuadas de acordo com a equação a seguir:

Pontuação de Orçamento =
$$\left(1 - \frac{Valor\ gasto\ no\ dia}{Orçamento\ etapa}\right) * 3,33$$
 (2)

Pontuação pelo controle do Tempo: Essa pontuação fornece reforço positivo aos jogadores de acordo com o tempo gasto para realização das atividades de cada etapa, de acordo com a expressão a seguir:

Pontuação de Tempo =
$$\left(1 - \frac{Dia\ Atual - Dia\ de\ Início}{Quantidade\ máxima\ de\ dias}\right) * 3,33$$
 (3)

Pontuação pela Produção: O SuperRisk avalia a eficiência da relação entre a produção alcançada pelo jogador e a meta definida pelo administrador do

jogo. O cálculo é baseado na quantidade de linhas de códigos produzidas. A expressão é dada por:

Pontuação de Produção =
$$\frac{Rendimento}{Meta} * 3,34$$
 (4)

A Pontuação final dos jogadores é calculada somando-se as pontuações nos quesitos de controle de orçamento, controle de tempo e controle da produção, conforme as expressões descritas anteriormente.

A expressão (5) descreve o calculo final da pontuação dos jogadores.

$$Pontuação Total = Orçamento + Tempo + Produção$$
 (4)

Cada projeto tem seu *Ranking* com a finalidade de comparar a atuação de cada jogador, promovendo assim uma competitividade.

6.4.2 Cálculo de percentual de Aumento de Salário

Além de contratar os funcionários disponíveis, há a possibilidade do jogador empregar funcionário já contratado por outro jogador. Para isso o jogador deve fazer uma solicitação de contratação, e por meio de um cálculo, o funcionário define seu novo valor, simulando uma negociação salarial. O jogador tem a opção de aceitar ou não a proposta. O cálculo da negociação é realizado baseado nos atributos de cada funcionário. O Cálculo é feito da seguinte forma:

% Aumento =
$$\frac{5 * V + 5 * M + 5 * TE + 40 * E + 5 * C + 40 * P}{600}$$
 (5)

Onde,

V é o valor do vigor do funcionário;

M é o valor da motivação do funcionário;

TE é o valor do trabalho em equipe do funcionário;

E é o valor da experiência do funcionário;

C é o valor da concentração do funcionário;

P é o valor da produtividade do funcionário.

A expressão (5) ilustra o percentual que o funcionário terá. Cada atributo do funcionário tem um peso para o cálculo. Os valores referentes à Experiência e Produção têm um peso maior, simulando assim a valorização dos funcionários mais produtivos e experientes.

Após calculada a porcentagem de aumento salarial, o valor do novo salário deve ser calculado. Esse procedimento é mostrado na equação (6).

Novo Salário =
$$\left(1 + \frac{\% Aumento}{100}\right) * Salário Antigo$$
 (6)

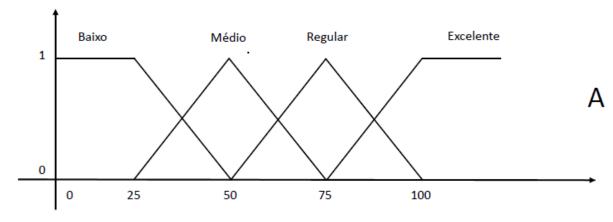
6.4.3 Sistema Fuzzy para afastamento dos funcionários

O Sistema Fuzzy Baseado em Regras é responsável por analisar o estado de saúde dos profissionais contratados e definir afastamentos temporários por licença médica. O sistema fuzzy baseado em regras faz parte de um agente reativo simples que atua no monitoramento dos atributos de motivação e vigor de todos os profissionais alocados em algum projeto. Assim, esse agente é capaz de determinar por quantos dias o profissional deve ser afastado em caso de doença.

A Figura 12 - (A) mostra os atributos de entrada do sistema: Motivação e Vigor e (B), a saída do sistema: Saúde do trabalhador.

Figura 12A – Gráficos de Fuzzyficação e Defuzzyficação.

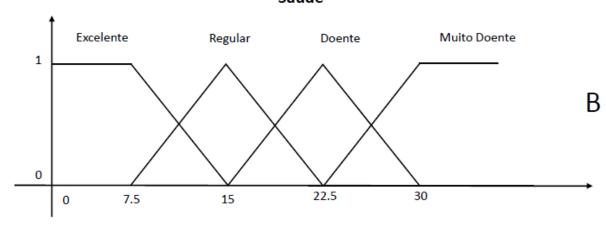
Motivação e Vigor



FONTE: Autoria Própria (2013).

Figura 13B – Gráficos de Fuzzyficação e Defuzzyficação.

Saúde



FONTE: Autoria Própria (2013).

Os atributos dos funcionários são representados por quatro conjuntos fuzzy, dois trapezoidais e dois em forma triangular. O domínio das variáveis de entrada está definido no intervalo [0,100] e contém os valores linguísticos difusos: *Baixo, Médio, Regular, e Excelente*. O domínio da saída do sistema representa o número de dias que o funcionário, supostamente, deve permanecer em casa e os valores linguísticos difusos: *Muito doente, Doente, Regular e Excelente*. Neste caso, o sistema fuzzy define a ausência de trabalho que pode ser de 0 a 30 dias que é inserido no atributo "Dias de Afastamento" do funcionário. O método defuzificação utilizado é o centro de gravidade.

Como a degradação dos atributos dos funcionários se dá de forma contínua, se não houvesse um fator de probabilidade associado aos agentes, assim que o vigor e motivação de algum fosse calculada abaixo de 90, o sistema fuzzy seria acionado e o funcionário já seria afastado por 3 dias, impedindo que seu estado físico continuasse a agravar até que fosse identificada uma situação mais crítica. Nesse caso, a probabilidade associada é um valor muito baixo, o que evita a ocorrência de muitos afastamentos e possibilita que a quantidade de dias na licença seja diferente em muitos casos.

O algoritmo do sistema fuzzy consiste em diferentes passos: as variáveis de entrada de vigor e motivação, como escalares, são transformados em valores linguísticos difuso pelas funções de fuzificação; esta informação é passada ao sistema fuzzy; os valores linguísticos difuso são transformados nos valores escalares da variável de saída através do processo de desfuzificação.

A Tabela 2 apresenta o mapa de regras da base de regras fuzzy do sistema fuzzy associando os conjuntos de entrada com as respectivas saídas resultantes. Foram criados quatro níveis de distinção entre as variáveis de saída, chamada de saúde (Excelente, Regular, Doente e Muito Doente).

Tabela 2 – Regras do sistema fuzzy para controle do afastamento por licença.

Motivação	Vigor	Saúde
Baixo	Baixo	Muito Doente
Baixo	Médio	Doente
Baixo	Regular	Regular
Baixo	Excelente	Regular
Médio	Baixo	Muito Doente
Médio	Médio	Doente
Médio	Regular	Regular
Médio	Excelente	Regular
Regular	Baixo	Doente
Regular	Médio	Regular
Regular	Regular	Regular
Regular	Excelente	Excelente
Excelente	Baixo	Regular
Excelente	Médio	Regular
Excelente	Regular	Excelente
Excelente	Excelente	Excelente

Fonte: Autoria Própria (2013).

Dessa forma, se o funcionário possuir motivação "baixa" e o vigor "médio", sua saúde é considerada "doente", todas essas regras são apresentadas na Tabela 2.

No SuperRisk jogador é informado sobre a ocorrência do imprevisto por meio do módulo de acompanhamento, detalhado no Anexo A. Durante o tempo de afastamento do funcionário, o jogador não conta com a produtividade diária desse funcionário na equipe de trabalho, porém o custo diário do mesmo continua a ser financiado. Outra característica importante neste cenário é que a opção de demiti-lo fica indisponível, impossibilitando o jogador dispensar um funcionário afastado.

6.5 VALIDAÇÃO DO SUPERRISK

Devido a restrições de tempo e turmas relativamente pequenas, o SuperRisk foi usado e avaliado por 21 alunos, 13 do Curso Técnico Integrado em Informática do Instituto Federal de Educação Ciências e Tecnologia do Rio Grande do Norte no campus localizado em Ipanguaçu, RN e 8 alunos do curso de Graduação em Ciência da Computação da Universidade Federal Rural do Semi-Árido, localizada em Mossoró, RN. Novas validações já foram agendadas e esses novos resultados serão apresentados em publicações futuras.

O *SuperRisk* se encontra completamente operacional, porém sido rodado em *localhost* tanto por limitações de configurações dos servidores da universidade quanto pela maior facilidade de correção de algum erro que venha acontecer no ato do processo validação. Um vídeo com o demonstrativo do uso da ferramenta está disponibilizado no blog do *SuperRisk*².

De acordo com (SAVI, 2011) para um jogo educacional ser considerado aplicável, seu uso deve ser rápido e simples, ou seja, não demandar conhecimentos avançados de nenhuma área e não possuir passos complexos. Diante disso, as informações coletadas na validação precisam ser consideradas significativas e coerentes.

- 1. A validação do SuperRisk seguiu a seguinte sequência:
- Preparação do formulário de avaliação;
- 3. Aplicação do questionário de avaliação de conhecimento prévio;
- 4. Apresentação do funcionamento e as regras do SuperRiskaos alunos;
- 5. Aplicação do questionário de avaliação do SuperRisk;
- 6. Discussão com as turmas:
- 7. Análise dos resultados.

A fim de poder avaliar o conhecimento dos alunos sobre o tema e também a aceitação dos mesmos em relação ao *SuperRisk*, foram preparados dois questionários.

_

² Link de acesso do blog do SuperRisk: http://superriskgame.blogspot.com.br/

O primeiro questionário foi aplicado antes dos alunos jogarem e objetivava avaliar o conhecimento prévio sobre o tema e também a experiência dos mesmos no âmbito de gerência.

O segundo questionário foi aplicado logo após os alunos fazerem uso da ferramenta e objetivava avaliar a satisfação dos alunos com relação ao jogo, o aprendizado dos mesmos e motivava os alunos a sugerirem modificações e adaptações para melhoria do jogo.

Após a primeira fase de validação do *SuperRisk* os formulários foram modificados. Todos os questionários se encontram no Anexo B dessa dissertação.

O cenário usado durante a validação foi o de uma empresa de confecção de roupas que solicitou o desenvolvimento de um software para otimizar a produção. O valor acordado pela empresa de confecção e a de desenvolvimento de software foi de 90 mil reais. O projeto foi dividido em 3 etapas, como demonstra a Tabela 3:

- Especificação com orçamento de 30 mil reais, produção de 6 mil artefatos, prazo de 10 dias para finalização e um nível fácil de dificuldade;
- Codificação com orçamento de 40 mil reais, produção de 12 mil linhas de código, prazo de 10 dias e nível difícil de dificuldade;
- 3. **Teste** com orçamento de 20 mil reais, produção de 12 mil linhas de correção de códigos, prazo de 5 dias e nível médio de dificuldade.

Levando em consideração que para a validação de cada turma o tempo disponível era de cerca de 1 hora, o *SuperRisk* foi configurado para que cada minuto no mundo real fosse equivalente a 1 dia no jogo. Dessa forma, para o cenário apresentado, os jogadores permaneceram jogando por 25 minutos.

No transcorrer da partida, os jogadores se mostraram bastante atentos e participativos, questionando, por exemplo, a falta de profissionais à disposição do mercado e a concorrência dos mais produtivos.

No final do jogo, todos comentaram sobre a pontuação de todos e trocaram experiência de como haviam chegado àquela pontuação.

Os alunos participantes da pesquisa responderam o questionário de avaliação do *SuperRisk* após utilizarem o jogo. O questionário está listado como Anexo dessa dissertação.

Tabela 3 – Cenário da validação

Etapas	Orçamento	Produção	Tempo (dia)
Especificação	R\$ 30.000,00	6.000 Artefatos	10
Codificação	R\$ 40.000,00	12.000 Linhas de código	10
Testes	R\$ 20.000,00	12.000 Linhas de código corrigidas	5
TOTAL	R\$ 90.000,00		25

Fonte: Autoria Própria (2014).

Algumas fotografias foram feitas com objetivo de registrar a pesquisa. Essas fotografias estão listadas no Anexo C.

Os resultados da validação do SuperRisk são apresentados a seguir.

6.5.1 Discussão sobre a avaliação dos alunos

Os resultados da validação do jogo são apresentados a seguir, divididos em subescalas que evolvem motivação, experiência do usuário e aprendizagem. No total, 21 alunos responderam ao questionário, sendo 13 alunos de curso técnico e 8 de curso de graduação.

Os alunos compõem uma comunidade de 21 componentes, sendo 66,7% do sexo masculino e 33,3% do sexo feminino, numa faixa etária de 16 à 23 anos, como apresentado na Tabela 4.

Tabela 4 – Sexo e Faixa etária dos avaliadores

SEXO Masculino Feminino 66,7% 33,3%				
Masculino Feminino				
66,7%	33,3%			
Faixa	Etária			
16 a 23 anos				

Fonte: Autoria Própria (2014).

De modo geral observa-se na Tabela 5, pelo Questionário de Conhecimento Prévio, que os alunos jogadores não têm muito conhecimento em gestão de riscos em projeto de softwares. 61,9% responderam que não possuíam nenhuma experiência, enquanto que 28,6% responderam que possuíam pouco

conhecimento e apenas 9,5% responderam que possuíam conhecimento razoável. Acrescentada a esta informação, foi constatado que 71,4% dos alunos que validaram a ferramenta nunca tiveram nenhum contato com a gerência de risco em projeto de software real. 23,8% tiveram pouca experiência e apenas 4,8% tiveram uma experiência razoável. Dessa forma a população avaliada possuía pouco conhecimento e consequentemente baixa experiência no assunto.

Tabela 5 – Resultado do questionário sobre a experiência prévia do jogador

·	Nenhum	Pouco	Razoável	Muito
Conhecimento em conceitos de gestão	61,9%	28,6%	9,5%	0%
de projeto de software	01,570	20,070	3,370	0 70
Experiência em Gestão de Projetos de	71.4%	23,8%	4.8%	0%
Software	11, 4 70	23,0%	4,070	U 70

Fonte: Autoria Própria (2014).

O Questionário de Avaliação do *SuperRisk* retrata uma circunstância onde é afirmado que o jogo foi considerado divertido e proporcionou uma experiência positiva aos alunos. A aparência do jogo foi considerada muito atrativa por 47,6% dos jogadores, razoável por 47,6% e pouco atrativa para 4,8%. Além disso, 81% dos alunos afirmaram que a forma do jogo ajudou a mantê-los atentos, atribuindo notas máximas, e 19% acharam que foi razoável a ajuda para manter a atenção.

O jogo foi considerado relevante para 57,1% dos alunos, que deram nota máxima para a adequação que o jogo tem para a facilidade do ensino da disciplina, seguido de 42,9% dos que acharam a ferramenta razoável para este mesmo quesito.

Outro item arguiu se o jogo é adequadamente desafiador. 85,7% responderam que a ferramenta é muito desafiadora, seguido de 14,3% que afirmaram que é razoavelmente instigante.

O item que pergunta se os alunos se divertiram com o jogo teve um nível de concordância de 76,2% afirmando que a ferramenta é muito divertida e 23,8% alegaram que a diversão proporcionada pelo jogo é razoável. Seguido a isto foi observado que o nível de concordância entre os alunos no quesito da competição entre os jogadores foi muito próximo do máximo, ou seja, 95,2% dos alunos que

validaram o *SuperRisk* atestaram que o jogo promove competição entre eles e 4,8% afirmaram que a competição é razoável.

Finalizando o Questionário de Avaliação do *SuperRisk*, foi perguntado se a ferramenta era suficientemente interessante a ponto de recomendarem. 85,7% afirmaram que recomendariam muito aos seus colegas seguidos de 14,3% que disseram que era razoável a recomendação aos amigos.

Tabela 6 – Resultado do questionário respondido para avaliação do jogo.

	Nenhum	Pouco	Razoável	Muito
A aparência do jogo é atrativa	0%	4,8%	47,6%	47,6%
A forma do jogo me manteve atento ao	0%	0%	19%	81%
jogo				
O jogo facilita meu aprendizado da	0%	0%	42,96%	57,1%
disciplina	• 70	0,0	.=,0070	01,170
O jogo proporcionou desafio	0%	0%	14,3%	85,7%
O jogo é divertido	0%	0%	23,8%	76,2%
O jogo promove competição entre os	0%	0%	4,8%	95,2%
jogadores	0 70	0 70	4,0 /0	33,Z 70
Eu recomendaria este jogo para meus	0%	0%	14,3%	85,7%
colegas	0 70	0 70	17,570	00,7 70

Fonte: Autoria Própria (2014).

7 CONSIDERAÇÕES FINAIS

A formação de profissionais qualificados em gerência de projetos de software requer o contato direto com os desafios e dificuldades relacionados à gerência de projetos. Devido às restrições de tempo e espaço inerentes aos cursos tradicionais de gerência de projetos, o uso de jogos torna-se uma opção prática de complementação da formação dos profissionais. Por meio de jogos é possível criar ambientes virtuais similares aos ambientes reais, simulando as dificuldades e desafios da tarefa de gerência de projetos, porém em um ambiente seguro e que pode ser acessado a qualquer momento.

É notório que os métodos tradicionais, como as aulas expositivas, apresentam deficiências especialmente no ensino de habilidades de aprendizagem menos teóricas e mais práticas.

Devido às vantagens dos elementos dos jogos, como a interação social, o desafio e a recompensa, o aluno tende a estar motivado e focado, facilitando com isso uma participação ativa nas atividades de ensino e aprendizado.

Nesse sentido nesse trabalho, foi modelado, implementado e validado um jogo persistente baseado em navegadores apoiado pelo uso de agentes inteligentes que monitora e controla o ambiente para alterar, de forma autônoma, as variáveis do sistema, afetando os profissionais, a organização e seus clientes.

Outro diferencial do *SuperRisk* é o uso de um sistema fuzzy baseado em regras para possibilitar a inserção de conhecimento específico da área de gestão de risco no sistema.

Uma das principais preocupações em relação ao desenvolvimento do jogo é a questão educacional. Assim é essencial que ao final do jogo o aluno tenha adquirido experiência prática na gestão de risco em projetos de softwares.

O SuperRisk foi usado e avaliado por 21 alunos, sendo 13 do Curso Técnico Integrado em Informática do Instituto Federal de Educação Ciências e Tecnologia do Rio Grande do Norte no campus localizado em Ipanguaçu, RN e 8 alunos do curso de Graduação em Ciência da Computação da Universidade Federal Rural do Semi-Árido, localizada em Mossoró, Rio Grande do Norte.

No transcorrer da validação, os alunos se mostraram bastante atentos e participativos, questionando problemas inerentes à gestão de projetos reais. No final

do jogo, os alunos comentaram sobre suas pontuações e trocaram experiência de como haviam finalizado as tarefas do *SuperRisk*.

7.1 PUBLICAÇÕES

Durante a realização deste trabalho, foi publicado o seguinte artigo em revista:

Revista RENOTE - Novas tecnologias na educação, nº 1, volume 11.
 ISSN: 1679-1916.

Também foram publicados dois artigos em congressos:

- XXII Ciclo de Palestras sobre Novas Tecnologias na Educação, evento ocorrido de 16 a 19 de junho;
- The 8th conference of the European Society for Fuzzy Logic and Technology, EUSFLAT-2013, evento ocorrido de 11 a 13 de setembro em Milão na Itália.

Também foi submetido um capítulo de livro sobre o trabalho para ser avaliado e, possivelmente incluído no livro "Fuzzy Approacher to e-Learning Systems", a ser publicado na série "Fuzziness and Soft Computing" da Springer - Publisher.

7.2 TRABALHOS FUTUROS

Durante o processo de validação foram recebidas sugestões para ajustes e adaptações do *SuperRisk*.

Por outro lado, algumas melhorias já haviam sido identificadas. Tais melhorias não foram implementadas por restrições de tempo.

Essas sugestões e melhorias incluem, basicamente:

- Melhorar a aparência do jogo para que as informações sejam exibidas de forma mais dinâmica:
- 2. Integrar o SuperRisk com redes sociais;
- 3. Facilitar a forma de configuração dos ciclos dos dias no jogo;
- 4. Desenvolver um tutorial explicativo;

- Aumentar a abrangência de gerenciamento de risco abordado pelo jogo;
- 6. Desenvolver uma versão para dispositivos móveis;
- 7. Ampliação da validação.

REFERÊNCIAS

AGARWAL, R.; UMPHRESS, D. A flexible model for simulation of software development process. In **48th Annual Southeast Regional Conference**, pp. 1-4. ACM, 2010.

AGILE. Lego scrum city. Disponível em:

http://www.agile42.com/en/training/scrumlego-city. Acesso em: 08 mai. 2013.

ALLUÉ, J. M. O grande livro dos jogos. [s.l.]. Leitura, 1999.

ALVES, E. M. S. P. Arquitetura pedagógica aplicada ao ensino de gerenciamento de projetos: a percepção do professor e dos alunos acerca da incorporação de um artefato tecnológico no planejamento didático. (2010). Dissertação de Mestrado, Mestrado Acadêmico em Computação Aplicada. UNIVALI, Itajaí, SC.

ÂNGELA FILIPAK MACHADO, C.; K. C. W. **Qualidade e produtividade em software.** Makron Books, 2001.

ATLEE, S. L. P. J. M. **Student study guide for software engineering**: theory and Practice. 4. ed. Prentice Hall, 2009.

AYAS, K.; ZENIUK, N. **Project-based learning**: building communities of reflective practitioners. Management Learning, 2003.

BECK, K. **Extreme programming explained**: embrace change. Addison-Wesley Professional, 2004.

BISHOP, C. M. Neural networks for pattern recognition. Claredon Press, 1995.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The unified modeling language user guide**. Pearson Education, 1999.

BREWER, J. L. Project managers: can we make them or just make them better? In: ACM, **6th Conference on Information Technology Education**, pp. 167-193, 2005.

CAMPOS, A., SIGNORETT, A.; LIMA, P.; LUIS, E.; FONTES, M.; DANTAS, K. Um jogo voltado à prática de gerenciamento de projetos. In: SBIE, Anais do XXII Simpósio Brasileiro de Informática na Educação, 2011.

CHADBOURNE, B. C. To the heart of risk management: teaching project teams to combat risk. In: **Annual Project Management institute seminar & symposium**, Pennsylvania, 1999.

CHAPTERS, P. Pmsurvey.org. 2013.

CHI, Z., YAN, H.; PHAM, T. Fuzzy algorithms with applications to image processing and pattern recognition. World Scientific, 1996.

- CHUNG, I. F., LIN, C.; LIN, C. A GA-based fuzzy adaptive learning control network. **Fuzzy Sets and Systems**, v. 1, p.65-84, 2000.
- CINTRA, M. E. Genetic generation of fuzzy rules with preselection of candidate rules. (2007). Dissertação de Mestrado, Federal University of São Carlos.
- CINTRA, M. E.; MONARD, M. C.; CAMARGO, H. A.; E MARTIN, T. P. Using fuzzy formal concepts in the genetic generation of fuzzy systems. In: **IEEE International Conference on Fuzzy Systems** (FUZZ-IEEE 2012), 2012. (in print).
- CINTRA, M. E.; NEVES, O. A. A fuzzy decision tree for bus network management. Proceedings of KDMiLe **1st Symposium on Knowledge Discovery**, Mining and Learning, n.1, p.1–8, 2013
- DANTAS, A. R.; BARROS, M. O.; E WERNER, C. M. L. A simulation-based game for project management experimental learning. In: **16th International Conference on Software Engineering and Knowledge Engineering**, pp. 19-24, 2004.
- OLIVEIRA, C. D. C.; CINTRA, M. E.; MENDES NETO, F. M. Learning risk management in software projects with a serious game based on intelligent agents and fuzzy systems. **8th Conference of the European Society for Fuzzy Logic and Technology**, pp. 834–839, 2013.
- DUDA, R. O.; HART, P. E. **Pattern classification and scene analysis**. John Wiley, 1973.
- DUMITRESCU, D.; LAZZERINI, B., E JAIR, L. Fuzzy sets and their application to clustering and training. Int. Series on Computational Intelligence. CBC Press, 2000.
- P. FILHO; PÁDUA, W. **Engenharia de software**: fundamentos, métodos e padrões. 2 ed. LTC, 2003
- GALVÃO, T. A. B.; MENDES NETO, F. M.; BONATES, M. F.; CAMPOS, M. T. A serious game for supporting training in risk management through projectbased learning, 2012.
- GONZALEZ, A.; PIREZ, R. SLAVE: A genetic learning system based on an iterative approach. **IEEE International Conference on Fuzzy Systems**, n. 7, p.176–191, 1999.
- HALL, E. **Managing risk**: methods for software systems development. Addison-Wes1ey Reading, 1998.
- HSU, R. C.; LIU, W. Project based learning as a pedagogical tool for embedded system education. In: **3rd International Conference on Information Technology**: Research and Education, 2005.

- ICHIBA, D.; HARA, K.; KANOH, H. Spatial interpolation of traffic data by genetic fuzzy system. **International Symposium on Evolving Fuzzy Systems**, n. 1, p.280-285, 2006.
- ITU, T. D. S. Measuring the information society 2010 / bdt. **Geneva**: International Telecommunication Union, 2010.
- JAIN, L. C.; MARTIN, N. M. Fusion of neural networks, fuzzy sets, and genetic algorithms: industrial applications. **The CRC Press International Series on Computational Intelligence**. 1998.
- JANG, S. R., SUN, C. T.; MIZUTANI, E. **Neuro-fuzzy and soft computing**. Prentice Hall, 1997.
- KECMAN, V. Learning and soft computing. MIT Press, 2001.
- KIELING, E.; ROSA, R. **Planager**: um jogo para apoio ao ensino de conceitos de gerência de projetos de software. (2006). Dissertação de Mestrado, Ciência da Computação. FACIN. Pontifícia Universidade Católica do Rio Grande do Sul.
- KLIR, G. J.; YUAN, B. **Fuzzy sets and fuzzy logic**: theory and applications. Prentice-Hall, 1995.
- LARSEN, P. M. Industrial applications of fuzzy logic control. **International Journal of Man-Machine Studies**, 12:3–10, 1980.
- LEVY, S. **Hackers**: heroes of the computer revolution. Anchor Press/Doubleday, Garden City, N.Y., 1984.
- LUKASIEWICZ, J. Philosophical remarks on many-valued systems of Propositional Logic. In: **Studies in Logic and the Foundation of Mathematics**, pp. 153-178, 1970.
- MAGDALENA, L.; MONASTERIO, F. A fuzzy logic controller with learning through the evolution of its knowledge base. **International Journal of Approximate Reasoning**, 16: 335-358, 1997.
- MAMDANI, E. H. Application of fuzzy logic to approximate reasoning using linguistic systems. **Fuzzy Sets and Systems**, n. 26, p.1182-1191, 1977.
- MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International Journal of Man-Machine Studies**, v. 7, n.1, p.135–147.
- MCGONIGAL, J. **Gaming can make a better world**. 2011. Disponível em: http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html Acesso em: 08 fev. 2013.

MONTEIRO, E. Nativos digitais já estão dominando o mundo e transformando a forma como o ser humano se comunica. Reportagem de O Globo. Disponível em: http://goo.gl/qfcln. 2011. Acesso em: 12 mai. 2011.

NAVARRO, E. O.; van DER HOEK, A. Simse: An interactive simulation game for software engineering education. In: **7th IASTED International Conference on Computers and Advanced Technology in Education**. 2004.

NICOLETTI, M. C.; CAMARGO, H. A. **Fundamentos da teoria de conjuntos fuzzy**. Série Apontamentos. EdUFSCar. 2004.

ONLINE OCIGRUP, S. **Browser games list**. 2013. Disponível em: http://www.browsergameslist.com/. Acesso em: 18 jan. 2013.

OULD, M. A. **Managing software quality and business ris**k. John Wiley & Sons. 1999.

PALUDO, L. **Rskmanager**: um jogo educativo de gerenciamento de riscos em projetos de software. (2013). Dissertação de Mestrado, Universidade do Vale do Itajaí curso de mestrado acadêmico em computação aplicada.

PEDRYCZ, W. Computational intelligence: an introduction. CRC Press, 1998.

PEDRYCZ, W.; GOMIDE, F. An introduction to fuzzy sets. MIT Press, 1998.

PIVEC, M.; DZIABENKO, O.; SCHINNERL, I. Aspects of game: based learning. In International Conference on Knowledge Management, pp. 217-224, 2003.

PIVETTA, V. U. **Modelo de apoio à gestão de riscos no desenvolvimento de software**. (2002). Dissertação de Mestrado, Programa de MBA em Engenharia de Software, Escola Politécnica da Universidade de São Paulo.

PMI, P. M. I. A guide to the project management body of knowledge. 3. ed. Newtown Square: Project Management Institute, 2004.

PONTES, A. A., MENDES NETO, F. M.; CAMPOS, G. A. L. Multiagent system for detecting passive students in problem-based learning. Ad. In: **Soft Comp.**, v. 71, pp. 165-172, 2010.

PRENSKY, M. **Digital natives**: digital immigrants – part 1. Washington: Mcb Up Ltd., 2001.

RUSSELL, S.; NORVIG, P. **Artificial intelligence**: a modern approach. 2. ed. Prentice Hall. 2004.

SANCHEZ, E.; SHIBATA, T.; ZADEH, L. A. **Genetic algorithms and fuzzy logic systems**. World Scientific, 1997.

SARKAR, S. Farmville 2 has 40 million monthly active users, amid other staggering numbers. 2014. Disponível em:

- http://www.polygon.com/2013/1/4/3837236/farmville-2-infographic-40-million-monthly-active-users. Acesso em: 10 jan. 2014.
- SAVI, R. **Avaliação de jogos voltados para a disseminação do conhecimento**. (2011). Tese de Doutorado, Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento. Florianópolis, SC.
- SCHWABER, K.; BEEDLE, M. **Agile software development with scrum**. Prentice Hal, 2001.
- SCHWALBE, K. **Information technology project management**. 6. ed. Cambridge: Course Technology Ptr. 2010. 400 p.
- SHAHZAD, S. Learning from experience: The analysis of an extreme programming process. In: **International Conference on Information Technology**: New Generations, 2009. pp. 1405-1410.
- SILVA, T. G.; BERNARDI, G.; MÜLLER, F. M. Abordagem de apoio ao ensino e aprendizagem de teste de software baseada em jogos sérios e mundos virtuais. In: Anais do **XXII Simpósio Brasileiro de Informática na Educação**, SBIE, 2011.
- SOMMERVILLE, I. **Software engineering**. Addison-Wesley Longman Publishing Co., Inc. 2006.
- SØRENSEN, B. H.; MEYER, B. Serious games in language learning and teaching: a theoretical perspective. In: **Digital Games Research Association International Conference**, 2007.
- TAKAGI, T.; SUGENO, M. Derivation of fuzzy control rules from human operator's control actions. In **Proceedings of the IFAC Symposium on Fuzzy Information**, **Knowledge Representation and Decision Analysis**, pp 55–60, 1983.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its application to modeling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 15, n. 1, pp. 116-132. 1985.
- TARAN, G. Using games in software engineering education to teach risk management. In **Proc. 20th Conf. Software Eng. Education and Training**, IEEE CS Press, 2007.
- TEIXEIRA., J.; SÁ, E.; FERNANDES, C. T. Representação de jogos educacionais a partir do modelo de objetos de aprendizagem. In: **XIII Workshop de Informática na Escola**, Rio de Janeiro, 2007.
- TEIXEIRA JUNIOR, C. A.; VILLAREAL, E. R. L.; CINTRA, M. E.; LIMA, N. W. B. Proposal of a fuzzy control system for the management of traffig lights. **11th Workshop on Intelligent Manufacturing Systems**, n. 1, pp. 243-248, 2013.

TORREAO, P. G. B. C. **Project management knowleadge learning environment**: ambiente inteligente de aprendizado para educação em gerenciamento de projetos. (2005). Dissertação de Mestrado, Centro de Informática. Universidade Federal de Pernambuco.

VINK, H. J. **Desdemona**: the next generation in movement simulation. 2012. Disponível em: http://goo.gl/FgyOp. Acesso em: 24 fev. 2013.

von WANGENHEIM, C. G.; RAUSIS, B. Z.; SOARES, G. M. Project detective. 2011.

WAGNER, R. W. Edgar dale: professional. **Theory Into Practice**, n. 9: pp.89-95., 1970.

WILSON, W. **The serious games initiative**. International Center for Scholars. Disponível em: http://www.seriousgames.org/. Acesso em: 21 jul. 2011.

XIAHOU, S. S. J. Integrated project based learning in software engineering education. In **International Conference on Educational and Network Technology**, 2010.

ZADEH, L. Outline of a new approach to the analysis of complex systems and decision process. **IEEE Transactions on Systems, Man and Cybernetics**, n. 3, p.28-44, 1973.

ZADEH, L. A. Fuzzy sets. Information and Control, n.8, p.338-353, 1965.

ZIMMERMANN, H. J. **Fuzzy set theory and its applications**. Kluwer Academic Publishers, 1991.

ZYNGA. **Farmville**: jogos no facebook. 2011. Disponível em: http://zynga.com/game/farmville-two. Acesso em: 18 Jan. 2013.

Anexo A – Detalhes Técnicos da Implementação do SUPERRISK

Neste anexo serão apresentados os detalhes técnicos da implementação do *SuperRisk*. Para a implementação do *SuperRisk* foram usadas tecnologias para internet: o sistema foi desenvolvido em *PHP*, o banco de dados em *MySQL* e os agente inteligentes foram desenvolvidos em JAVA, com o *framework* JADE 4.3.

A.1 VISÃO GERAL E MÓDULOS DO SUPERRISK

No SuperRisk, seus módulos interagem entre si para simular um ambiente real de gerência. Os módulos foram desenvolvidos para simular os papeis de jogador e de administrador da ferramenta. Basicamente, o administrador é o professor que controla todos os parâmetros do jogo, além da participação dos alunos.

A.1.1 Módulo de Cadastro de Novo Jogador

O cadastro no *SuperRisk* é pré-requisito para acesso ao sistema. A partir deste cadastro é possível que o administrador do jogo aloque jogadores a projetos e que os jogadores acessem o ambiente. O cadastramento é realizado pelo usuário na opção "*Não* é cadastrado? Clique aqui." (Ver Figura 14).

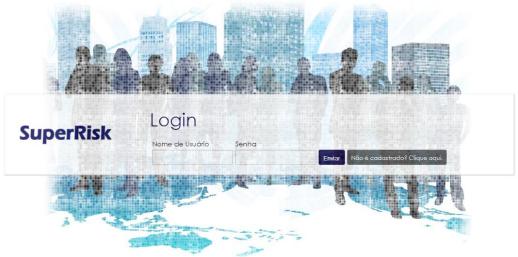


Figura 14 – Tela de Login do SUPERRISK

Fonte: Autoria própria (2014)

Apenas informações básicas, todas obrigatórias, são necessárias para o cadastro. Caso uma ou mais dessas não seja inserida, uma mensagem de erro é exibida.

Após a etapa de cadastro, o jogador pode acessar o sistema. Inicialmente o jogador não tem acesso a nenhum projeto. Caso o jogador precise alterar seus dados de cadastro, é possível alterar essas informações em "*Meus Dados*" (Ver Figura 15).



Figura 15 – Tela de edição de cadastro

Fonte: Autoria própria (2014)

Para que o jogador tenha acesso aos projetos, é necessário que o administrador do sistema aloque-o a algum projeto. Após ter acesso ao projeto, o jogador tem acesso a todos os módulos que controlam o projeto.

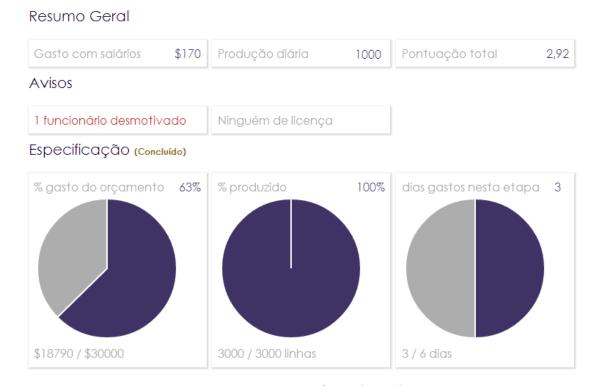
A.1.2. Módulo de Acompanhamento de Projeto

Esta funcionalidade tem como objetivo a visualização do andamento do projeto. Ela permite ao jogador acesso às seguintes informações:

- 1. Otimizar a quantidade de dias restantes para a etapa finalizar;
- 2. O total do orçamento gasto em cada etapa;
- 3. Valor diário comprometido com salário de funcionários contratados;

- 4. Número de funcionários desmotivados ou de licença médica;
- 5. Sua pontuação, permitindo a competição entre os jogadores;
- 6. A quantidade de linhas produzidas até então, como mostra a Figura 16.

Figura 16 – Tela de acompanhamento do projeto. Acompanhamento



Fonte: Autoria própria (2014)

Neste ambiente o jogador também é informado sobre a ocorrência de imprevistos do jogo por meio de mensagens instantâneas, além de outros informes referentes ao jogo. Estas mensagens são fundamentais para manter o jogador atualizado quanto às mudanças ocorridas no projeto e constitui uma das principais ferramentas de comunicação do jogo.

A.1.3. Painel de pontuação dos jogadores

Esse modelo tem o objetivo de promover a concorrência entre os jogadores. Nele é possível visualizar e comparar a pontuação de cada um dos jogadores de um determinado projeto, como ilustrado na Figura 17.

Figura 17 – Tela de classificação dos jogadores

Ranking

1° Ester

8.85
Pontos

2° igo

3° Isabelle

Ranking

8.85
Pontos

3.53
Pontos

Fonte: Autoria própria (2014)

A.1.4. Módulo de Mercado de Trabalho

Liderança

Para o jogador começar o desenvolvimento de um projeto, ele precisa de recursos humanos. Assim, ele deve contratar funcionários, este módulo é responsável por essa funcionalidade. Aqui são listados os profissionais disponíveis. Esses profissionais possuem atributos que permitem que o gerente os avalie antes da contratação. Estes atributos são:

Nome	União de Nome e Sobrenome que identificará o
	profissional;
Custo Diário	Custo diário de cada profissional definido em
	dólares;
Produtividade	Valor da produtividade diária de cada funcionário
	definido em linhas de código;
Trabalho em Equipe	Valor referente à característica de trabalho em
	equipe do profissional;
Motivação	Valor que indica o estado de motivação de cada
	funcionário;
Vigor	Valor usado para quantificar o estado de saúde do
	funcionário.

Valor referente à liderança;

Concentração Valor referente à concentração do funcionário;

Experiência Valor referente à experiência do funcionário;

Dias de Afastamento Quantidade de dias que o funcionário está de

licença médica.

Também é possível visualizar os funcionários contratados por outros jogadores, para simular a competição existente no mundo real. Assim os jogadores podem aceitar a proposta feita pelo funcionário no ato da contratação.

Figura 18 – Tela com o resumo do Mercado de Trabalho disponível no projeto para todos os jogadores



Fonte: Autoria própria (2014)

A.1.5. Módulo de Equipe de Trabalho

Após a contratação dos funcionários no mercado de trabalho, o jogador tem acesso às informações de sua equipe, com mostra a Figura 19. Neste ambiente é possível visualizar dinamicamente os atributos de cada funcionário e assim tomar medidas para gerenciar a produtividade de sua equipe. Caso o jogador não esteja satisfeito com o desempenho de um dos funcionários, neste módulo também é

possível demiti-lo, exceto em caso de ocorrência de licença médica para o funcionário em questão. Essa funcionalidade também permite que o jogador acompanhe o tempo restante dos dias de licença de cada um dos funcionários afastados.

Figura 19 – Tela que mostra a Equipe do jogador Home » Projeto de Avaliação » Minha Equipe Minha Equipe Contratado Danilo Santos Produtividade: 108 Custo Diário: R\$ 78,00 Concentração: 46
Motivação: 100 Vigor: 100 Experiência: 25
Licença Saúde: - Trabalho em Equipe: 22 Demitir Contratado Cláudio Santos Produtividade: 25 Custo Diário: R\$ 93,00 Concentração: 50
Motivação: 100 Vigor: 100 Experiência: 17
Licença Saúde: - Trabalho em Equipe: 81 Demitir Contratado Francisco Pereira Custo Diário: R\$ 49,00 Concentração: 75 Vigor: 100 Experiência: 18 Produtividade: 25 Demitir Motivação: 100 Licença Saúde: -Trabalho em Equipe: 4

Fonte: Autoria própria (2014)

A.1.6. Módulo de Aquisições

Uma restrição imposta pelo \erg para contratação de novos funcionários é a compra de novos computadores para cada um deles. No módulo de aquisições (Figura 20), além da funcionalidade citada anteriormente, também é possível comprar os seguintes produtos:

- 1. Livros;
- 2. Palestras motivacionais;
- 3. Melhorias na internet;
- Impressoras;
- 5. Máquina de café;
- 6. Atividades cooperativas;
- 7. Curso:
- 8. Hora Extra;
- 9. Licença de Software;

10. Ferramentas.

Cada um desses produtos tem características que influenciam os atributos dos funcionários.

Figura 20 – Tela de Aquisições do projeto Home » Projeto de Avaliação » Aquisições **Aquisições** Minhas Aquisições Computador 3 Adquirir Mais Itens Clique nos itens para comprá-los Atividade Cooperativa \$2.000 \$4.000 Computador Produtividade +10 Motivação +10 📫 Curso \$6.000 >_ Ferramenta Case \$3.500 Produtividade +10 Vigor +5 \$3.750 impressora \$800 (I) Hora Extra Produtividade +15 Motivação -20 Produtividade +10 Experiência +10 Concentração +10 Licença de Software \$300 \$2.500 Livro Produtividade +10 Vigor -10

Fonte: Autoria própria (2014)

A.1.7. Módulo de Eventos

Todas as ações que os jogadores fazem no jogo são salvas neste módulo. As informações são organizadas por dia para que o jogador acompanhe a evolução do projeto (ou projetos) sob sua gerência (ver Figura 21).

Figura 21 – Tela do Módulo de registro dos eventos do jogador no decorrer do projeto

Fonte: Autoria própria (2014)

A.1.8. Módulo de Fórum

O SuperRisk também possui um ambiente que permite a troca de informações entre os jogadores, chamado de fórum (Figura 22). No Fórum, os jogadores podem se comunicar entre si e com o administrador do jogo, que pode ser o professor da disciplina, com o intuito de tirar as dúvidas e compartilhar experiências.

Figura 22 – Fórum do jogo

Home » Projeto de Avaliação » Fóruns » Tópico: Dúvida

Tópico: Dúvida

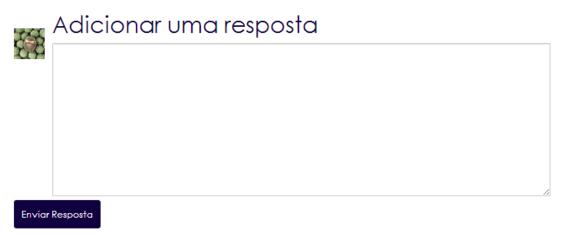
Tópico adicionado com sucesso



Data e hora da postagem: 10/01/14 01:03

É melhor contratar experientes ou treiná-los?

Danilo Câmara



Fonte: Autoria própria (2014)

A seguir são descritos os módulos de uso exclusivo do administrador.

A.1.9. Módulo para Criação de Projetos

O Administrador cadastra novos projetos nesse módulo. Para criar um projeto, o administrador deve informar o nome para o projeto, uma descrição que será visível aos jogadores, um identificador e a data e hora de inicio do projeto, como ilustrado na Figura 23.

Cada projeto é dividido em etapas e a maneira como o administrador adicionará essas etapas é descrita a seguir.

Formato HH:MM

Figura 23 – Tela de Criar Projeto

Home » Criar Projeto

Criar Projeto

- Após o início do projeto, não será possível a alteração de nenhuma informação desta página.
- Após alocar usuários neste projeto, não será possível incluir ou retirar etapas, somente alterá-las.

Dados do projeto

Nome do Projeto

Projeto de Avaliação

Descrição do Projeto

Projeto de Avaliação dos alunos da turma de informática.

Base da URL

Data de Início

Hora de Início

14/02/2014

Fonte: Autoria própria (2014)

Formato DD/MM/AAAA

A.1.10. Módulo para criar Etapas

Precisa ser um nome curto, em minúsculas e sem espaços. Esta

informação será a base para a url do projeto.

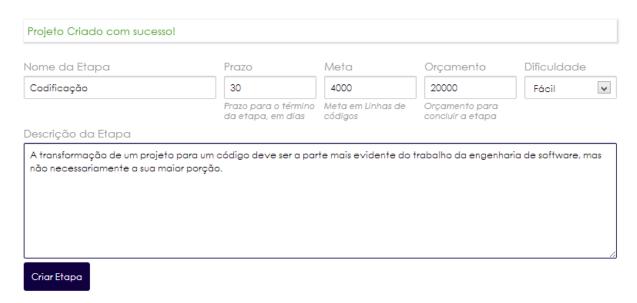
Criar Projeto

Após a criação dos projetos, o administrador deve adicionar as etapas dos mesmos neste módulo (ver Figura 24). Cada etapa tem suas informações características, como nome, tempo de duração (prazo de término de cada etapa), meta de produção (definida em linhas de código), orçamento, grau de dificuldade e a descrição que fica disponível aos jogadores. Um projeto pode ter a quantidade de etapas que o administrador achar necessária.

Figura 24 – Tela do módulo para criar Etapa

Home » Projeto de Avaliação » Editar Projeto » Adicionar Etapas

Adicionar Etapas



Fonte: Autoria própria (2014)

A.1.11. Módulo para alocar Jogadores

Após a criação do projeto e das etapas, o administrador do jogo terá a opção de escolher quais jogadores, já cadastrados, participarão do projeto. Essa funcionalidade permite que administrador controle a quantidade de jogadores por projeto. A Figura 25 apresenta a tela do módulo.

Figura 25 - Funcionalidade para alocar jogadores ao projeto

Home » Projeto de Avaliação » Alocar usuários

Alocar usuários

Após alocar usuários, não será possível incluir e/ou remover etapas deste projeto

Clique nos itens para alocar e desalocar os usuários deste projeto

Dr. Sheldon Cooper

Cláudio Medeiros

Maria Medeiros

Danilo Câmara

Marcos Evandro

Cintra

Eliseu

Fonte: Autoria própria (2014)

A.1.12. Módulo de configurar Aquisições

Com a finalidade de tornar o \erg mais dinâmico, o administrador pode configurar, através deste módulo, os valores de atuação que cada uma das aquisições tem nos funcionários dos projetos, como mostra a Figura 26. Além disso, também é possível adicionar novos produtos e alterar seus valores.

Figura 26 – Tela de configuração das aquisições

Home » Configurações » Aquisições

Aquisições

ld	Name	Cost	Effect	Productivity	Motivation	Experience	Teamwork	Vitality	Concentration	Edit
1	Computador	\$4.000		10	10	-	-	-	-	
2	Upgrade de Internet	\$3.000		15	-	-	-	-	-	
3	Impressora	\$800		10	-	10	-	-	10	
4	Livro	\$300		10	-	-	-	-10	-	
5	Curso	\$6.000		10	-	-	-	5	-	
6	Palestra	\$2.000		-	-	-	15	10	-	
7	Palestra Motivacional	\$1.500		-	10	-	10	10	-	
8	Licença de Software	\$2.500		_	-	-	-	-	-	
9	Ferramenta Case	\$3.500		-	-	-	-	-	-	
10	Máquina de Café	\$200		10	5	-	-	-20	-	
11	Sistema de Backup	\$1.000		-	-	10	-	-	-	
12	Hora Extra	\$3.750		15	-20	-	-	-	-	
13	Atividade Cooperativa	\$2.000		-	-	-	-	-	-	

Fonte: Autoria própria (2014)

A.1.13. Módulo de acompanhar Jogadores

Semelhante ao módulo disponível aos jogadores, o administrador poderá acompanhar o desempenho de cada um dos jogadores na tela do módulo. Esse

módulo tem como objetivo expor as dificuldades enfrentadas pelos jogadores ao administrador. A partir daí o administrador pode agir para sanar algumas dúvidas.

Anexo B - Questionários

1 PRIMEIRO QUESTIC	ONARIO					
	SuperRisk					
Instituição de Ensino:	-					
Idade:	_ Sexo: ()M ()F					
Formação:						
	ser respondido antes do uso da ferramenta.		1	2	3	4
Experiência em Gestão de	itos de gestão de projeto de software	-			\vdash	
Experiencia em destas de	o i rojetos de sortware					
	1 – Nenhum / 2- Pouco / 3- Razoável / 4-Muito ser respondido após o uso da ferramenta.					
			1	2	3	4
A Aparência do jogo é ati	rativa					
A forma do jogo me man	teve atento ao jogo					

	1	2	3	4
A Aparência do jogo é atrativa				
A forma do jogo me manteve atento ao jogo				
O Jogo facilita meu aprendizado da disciplina				
O Jogo proporcionou desafio				
O jogo é divertido				
O jogo promove competição entre os jogadores				
Eu recomendaria este jogo para meus colegas				

1 – Não / 2- Pouco / 3- Razoável / 4-Muito

Observações adicionais		

B.2 QUESTIONÁRIO MODIFICADO

SuperRisk

	-				
Instituição de Ensino:					
Idade:S	Sexo: ()M ()F				
Formação:					
					_
1 -) Questionário 1, deverá ser re	espondido antes do uso da ferramenta.				
		1	2	3	
Conhecimento em conceitos de					
Experiência em Gestão de Proje	etos de Software				
					_
1 – Nenł	hum / 2- Pouco / 3- Razoável / 4-Muito				
Tem experiência prática em dese	envolvimento de software? (V) (F)				
Se sim					
() Até 6 meses;					
() De 6 meses a um ano);				
() De um ano a dois and	os;				
() Mais de dois anos.					
Tem experiência em controle de	gastos? (V) (F)				
Se sim					
() Até 6 meses;					
() De 6 meses a um ano	o;				
() De um ano a dois and	os;				
() Mais de dois anos.					
Já liderou equipes de trabalho (q	qualquer área)? (V) (F)				
Se sim					
() com 2 até 5 pessoas;					
() com 6 até 10 pessoas	5;				
() com 11 até 20 pessoa	as;				
() Mais de 20 pessoas					
Gosta de Jogos? (V) (F)					
Qual frequência do uso o	de jogos por mês:				

Já jogo	u algum jogo de simulação? (V) (F)				
	Qual:				_
- \ -					
2 -) Qu	estionário 2, deverá ser respondido após o uso da ferramenta.				
		1	2	3	4
	A Aparência do jogo é atrativa				
	A forma do jogo me manteve atento ao jogo				
	O Jogo facilita meu aprendizado da disciplina				
	O Jogo proporcionou desafio				
	O jogo é divertido				
	O jogo promove competição entre os jogadores				
	Eu recomendaria este jogo para meus colegas				
Suges	tões e melhorias				
Obse	rvações adicionais				

Anexo C - Registro da Validação

Figura 27 – Turma de validação. IFRN - Campus Ipanguaçu. 10/01/2014.



Fonte: Autoria própria (2014)

Figura 28 - Turma de validação. IFRN - Campus Ipanguaçu. 10/01/2014.



Fonte: Autoria própria (2014)



Figura 29 – Turma de validação. UFERSA. 13/01/2014.

Fonte: Autoria própria (2014)