

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO**

**ALOCAÇÃO DE PESSOAS EM PROJETOS DE
SOFTWARE UTILIZANDO SISTEMAS
MULTIAGENTES**

DISSERTAÇÃO DE MESTRADO

Liane Santiago Cafarate

Santa Maria, RS, Brasil

2011

ALOCAÇÃO DE PESSOAS EM PROJETOS DE SOFTWARE UTILIZANDO SISTEMAS MULTIAGENTES

por

Liane Santiago Cafarate

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Gerência da Produção, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Engenharia de Produção.**

Orientador: Prof. Dr. Marcos Cordeiro d'Ornellas

Santa Maria, RS, Brasil

2011

**Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**ALOCAÇÃO DE PESSOAS EM PROJETOS DE SOFTWARE
UTILIZANDO SISTEMAS MULTIAGENTES**

elaborada por
Liane Santiago Cafarate

como requisito parcial para obtenção do grau de
Mestre em Engenharia de Produção

COMISSÃO EXAMINADORA:

Marcos Cordeiro d'Ornellas, Dr.
(Presidente/Orientador)

Lisandra Manzoni Fontoura, Dr. (UFSM)

Rejane Frozza, Dr. (UNISC)

Santa Maria, 23 de Agosto de 2011.

Para o meu amor, Henrique Vicentini.

Agradecimentos

Agradeço primeiramente a Deus, por toda a força e apoio que ele me dá. Sem a Sua Vontade eu não teria chegado onde eu cheguei. Também agradeço aos meus pais, irmãos por sempre acreditarem no meu potencial.

Agradeço ao meu orientador Marcos Cordeiro d'Ornellas, pelo acompanhamento em praticamente toda a minha vida acadêmica, por todas as oportunidades que tive através dele, pelas conversas e amizade.

À Lisandra Manzoni Fontoura pela ajuda que ela me prestou durante o meu mestrado, pelos ensinamentos e amizade.

Ao meu eterno namorado, Henrique Vicentini, pela infinita ajuda em todas as áreas da minha vida. Por estar sempre ao meu lado em qualquer momento e me apoiar nas minhas decisões. Sem ele eu não seria tão feliz como eu sou hoje.

Ao meu amigo Daniel Michelin De Carli, pelas conversas úteis e incentivos que me fizeram continuar essa caminhada. Por todas as coisas que me ensinou em todos esses anos de faculdade.

Aos chefes que tive durante este período, por permitirem minhas viagens e entenderem a importância que o meu mestrado tinha para minha carreira.

À Capes pelo apoio financeiro.

“O primeiro requisito para o sucesso é a habilidade de aplicar incessantemente suas energias físicas e mental a qualquer problema, sem se cansar.”

(Thomas Edison)

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Maria

ALOCAÇÃO DE PESSOAS EM PROJETOS DE SOFTWARE UTILIZANDO SISTEMAS MULTIAGENTES

AUTORA: LIANE SANTIAGO CAFARATE

ORIENTADOR: DR. MARCOS CORDEIRO D'ORNELLAS

Data e Local da Defesa: Santa Maria, 23 de Agosto de 2011.

O processo de atribuição de tarefas é uma atividade chave na gerência de projetos. Ser capaz de conciliar o perfil das pessoas, bem como seus anseios profissionais nesse processo é um desafio para gestores. Projetos de software diferem dos demais devido a sua intangibilidade e complexidade, características que contribuem ao fracasso desses projetos. Além disso, tais projetos dependem de mão-de-obra altamente especializada e capacitada e saber motivá-las e desenvolvê-las é um fator de extrema importância para alcançar êxito nessa área. Dessa forma, o presente trabalho buscou apresentar um modelo de alocação de tarefas em projetos de software que considera, além dos objetivos organizacionais (custos, prazos, entre outros), os anseios profissionais dos membros participantes. A estratégia proposta para a investigação é através da simulação de sistemas multiagentes, sistemas geralmente utilizados para simulação de entidades comportamentais. Com esse modelo pretendeu-se prover ao gestor uma ferramenta de auxílio na tomada de decisão no processo de alocação de tarefas em empresas de desenvolvimento de software, permitindo a visualização de diversos cenários e estratégias, gerando dessa forma melhor desempenho na execução de projetos e maior satisfação dos profissionais.

Palavras-chave: gestão de projetos de software, sistemas multiagentes, gestão de pessoas, alocação de recursos humanos.

ABSTRACT

Master Dissertation
Production Engineering Graduate Program
Federal University of Santa Maria

PEOPLE SCHEDULING IN SOFTWARE PROJECTS USING MULTI-AGENTS SYSTEMS

AUTHOR: LIANE SANTIAGO CAFARATE

ADVISOR: DR. MARCOS CORDEIRO D'ORNELLAS

Date and Place of Dissertation Seminar: Santa Maria, 23th August, 2011.

The tasks allocation process is a key activity in project management. Being able to conciliate the people profile, as well their professional aspirations is a real challenge for managers. Software projects differ from the rest because of its intangibility and complexity, characteristics that contribute to the failure of these projects. In addition, such projects depend on skilled and highly specialized manpower and the knowledge to motivate and develop them is an extremely important factor for achieving success in this area. Thus, this document sought to present a model of task allocation in software projects that considers, in addition to organizational goals (cost, time, etc.), the professional aspirations of the participating members. The proposed strategy for research is through the simulation of multi-agent systems, systems generally used for behavioral simulation entities. Through this model was intended to provide the manager a tool to aid in decision making in the process of task allocation in software development companies, allowing them to view different scenarios and strategies, looking for producing a better performance in project implementation and greater people satisfaction.

Keywords: software project management, multi-agent systems, people management, human resource allocation.

LISTA DE TABELAS

Tabela 2.1 – Comparação entre Projeto, Programa e Portfolio.....	22
Tabela 2.2 – Exemplo de uma classificação de competência	31
Tabela 3.1 – Comparativo entre agentes reativos e cognitivos.....	41
Tabela 4.1 – Método da Pesquisa.....	47
Tabela 5.1 – Caso de Uso Interno “Controlar a alocação da Atividade”	55
Tabela 5.2 – Caso de Uso Interno “Calcular a duração do projeto”	55
Tabela 5.3 – Caso de Uso Interno “Calcular o custo do projeto”	56
Tabela 5.4 – Caso de Uso Interno “Buscar tarefa”	56
Tabela 5.5 – Caso de Uso Interno “Negociar tarefa”	57
Tabela 5.6 – Caso de Uso Interno “Executar tarefa”	58
Tabela 5.7 – Crenças dos agentes.....	63
Tabela 5.8 – Cartões de Agentes BDI.....	64
Tabela 5.9 – Elementos de uma atividade	66
Tabela 5.10 – Comparação de elementos de atividade	67
Tabela 5.11 – Elementos de uma Atividade.....	68
Tabela 5.12 – Atributos da Competência	69
Tabela 5.13 – Níveis de experiência	69
Tabela 5.14 – Atributos de um recurso humano.....	70
Tabela 5.15 – Comparação de atributos de recursos humanos.....	70
Tabela 5.16 – Atributos de uma Pessoa	71
Tabela 5.17 – Valores de experiência para evoluir nível.....	72
Tabela 5.18 – Níveis de conhecimento de competências	78
Tabela 5.19 – Plataformas para desenvolvimento de sistemas multiagentes	79
Tabela 6.1 – Configurações do Modelo.....	82
Tabela 6.2 – Configuração "Melhor Prazo"	83
Tabela 6.3 – Configuração "Melhor Custo"	83
Tabela 6.4 – Configuração "Aprendizado 30%"	83
Tabela 6.5 – Configuração "Aprendizado 100%"	84
Tabela 6.6 – Resumo Melhor Prazo.....	84
Tabela 6.7 – Resumo Melhor Custo.....	84
Tabela 6.8 – Resumo Aprendizado 30%.....	84
Tabela 6.9 – Resumo Aprendizado 100%.....	85
Tabela 6.10 – Resumo Total	85
Tabela 6.11 – Resultados ANOVA.....	85
Tabela 6.12 – Comparação de custos médios em porcentagem	86
Tabela 6.13 – Comparação de prazos médios (dias úteis) em porcentagem	87
Tabela 6.14 – Comparação entre pontos de experiência em porcentagem	88

Tabela 7.1 - Comparação entre trabalhos correlatos e essa dissertação	94
---	----

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Triângulo das Restrições.....	21
Figura 2.2 – Diagrama ilustrativo de Projetos, Programas e Portfolios	23
Figura 2.3 – Ciclo de vida do projeto	25
Figura 2.4 – Processo de gestão de pessoas em projetos.....	28
Figura 2.5 – Modelo de competência.	30
Figura 3.1 – Estrutura típica de sistemas multiagentes.....	40
Figura 3.2 – Modelo de Agentes BDI.	43
Figura 5.1 – Processo de modelagem do Framework BDI.....	51
Figura 5.2 – Caso de uso externo	52
Figura 5.3 – Hierarquia de objetivos - Gestor.....	53
Figura 5.4 – Hierarquia de objetivos – Membro.....	53
Figura 5.5 – Diagrama de Sequência Buscar Tarefas.....	59
Figura 5.6 – Diagrama de Sequência Negociar Tarefas	59
Figura 5.7 – Diagrama de Sequência Executar Tarefas.....	60
Figura 5.8 – Diagrama de Atividades de Agente	61
Figura 5.9 – Diagrama de Fluxo de Dados.....	62
Figura 5.10 – Diagrama de Classes Conceitual	64
Figura 5.11 – Cálculo da Experiência Ganha pelo Agente Membro.....	72
Figura 5.12 – Comportamento dos Agentes.....	74
Figura 5.13 – Comunicação Membro-Gestor	75
Figura 5.14 – Comunicação Membro-Membro	76
Figura 5.15 – Cálculo de Compatibilidade.....	77
Figura 5.16 – Algoritmo do Grau de Compatibilidade.....	78
Figura 5.17 – Fórmula de Compatibilidade do Método de Aprendizado.....	78
Figura 6.1 – Comparação entre custos médios.....	86
Figura 6.2 – Comparação entre prazos médios (dias úteis).....	87
Figura 6.3 – Comparação entre pontos de experiência.....	88
Figura 6.4 – Resumo das configurações.....	89

LISTA DE ABREVIATURAS E SIGLAS

BDI: Beliefs, Desires, Intentions

CMMI: Capability Maturity Model Integration

JADE: Java Agent Development Framework

MAS: Multi-Agent Systems

ROI: Return on Investment

UML: Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Caracterização do Problema	15
1.2 Contribuições da Pesquisa.....	17
1.3 Justificativa.....	18
1.4 Estrutura da Dissertação	19
2 ALOCAÇÃO DE PESSOAS EM PROJETOS DE SOFTWARE.....	20
2.1 Conceitos Relacionados a Projetos.....	20
2.2 Gerência de Projetos de Software	23
2.3 Processos de Gestão de Projetos.....	24
2.4 Gestão de Pessoas em Projetos	27
2.4.1 Identificação de Competências	29
2.4.2. Seleção de Pessoas.....	31
2.4.3. Monitoramento do Processo	32
2.4.4. Avaliação do Desempenho e Desenvolvimento da Equipe	33
2.5 Objetivos Individuais e Organizacionais.....	35
2.6 Considerações Finais	36
3. AGENTES DE SOFTWARE	38
3.1 Definição de Agentes Inteligentes e suas Características	38
3.2 Sistemas Multiagentes (MAS) e suas Classificações.....	39
3.3 Arquitetura de Sistemas Multiagentes	41
3.4 Aplicações de Sistemas Multiagentes.....	44
3.5 Considerações Finais	45
4 MÉTODO DE PESQUISA.....	46
4.1 Etapas do Desenvolvimento.....	47
5 CONSTRUÇÃO DO MODELO DE SIMULAÇÃO	49

5.1	Processo de Modelagem	49
5.2	Propósito do Modelo	50
5.3	Identificação dos Agentes	50
5.4	Modelagem BDI	50
5.4.1	Casos de Uso Externos	51
5.4.2	Hierarquia de Objetivos	52
5.4.3	Casos de Uso Internos	54
5.4.4	Diagramas de Sequência	58
5.4.5	Diagramas de Atividades de Agente	60
5.4.6	Lista de Crenças do Agente	60
5.4.7	Cartões de Agentes BDI	63
5.5	Premissas do Modelo	64
5.6	Plataforma para Sistemas Multiagentes	78
6	RESULTADOS EXPERIMENTAIS	80
6.1	Verificação do Programa de Simulação	80
6.2	Validação do Modelo	81
6.3	Experimentação do Modelo	81
6.4	Resultados Encontrados	82
7	CONSIDERAÇÕES FINAIS	90
7.1	Limitações do Estudo	91
7.2	Trabalhos Correlatos	92
7.3	Trabalhos Futuros	94
8	REFERÊNCIAS BIBLIOGRÁFICAS	95
	ANEXOS	106
	ANEXO A – DADOS DO PROJETO	107
	ANEXO B – DADOS DOS RECURSOS HUMANOS	108
	APÊNDICE A – DADOS DE APRENDIZAGEM	110
	APÊNDICE B – PLANEJAMENTO MELHOR PRAZO	111
	APÊNDICE C – PLANEJAMENTO MELHOR CUSTO	112
	APÊNDICE D – PLANEJAMENTO APRENDIZADO 30%	113
	APÊNDICE E – PLANEJAMENTO APRENDIZADO 100%	114

1 INTRODUÇÃO

Este capítulo introdutório trata da caracterização do problema estudado neste trabalho, assim como a justificativa do tema escolhido. Além disso, são apresentados os objetivos e contribuição da pesquisa, bem como a estrutura desta dissertação.

1.1 Caracterização do Problema

Desenvolver software é uma atividade de trabalho intensivo (ANTONIOLO, CIMITILE, *et al.*, 2004). Muitas organizações de software enfrentam dificuldades diárias para desenvolver aplicações com qualidade em tempo hábil, dentro do orçamento e que satisfaçam as necessidades de negócios (BAE, 2007).

Um relatório chamado CHAOS (STANDISH GROUP, 2009) apresenta resultados preocupantes em relação à realização de projetos de desenvolvimento de software. Nesse estudo, a taxa de projetos bem sucedidos em 2009 é de 32%, decrescente em relação ao ano de 2006 (35%). Ainda, 44% dos projetos tiveram mudanças (atrasos, custos acima do planejado e/ou reduções de escopo) e 24% dos projetos falharam (foram cancelados antes de sua conclusão ou entregues e nunca usados) contra 19% de projetos falhos em 2006.

Entretanto, apesar de diversas pesquisas nessa área, a realização de projetos de software bem sucedidos ainda é uma grande dificuldade para as empresas. Muitos fatores impedem a conclusão desse tipo de projeto dentro do prazo e orçamento estabelecido, com as funcionalidades e o nível de qualidade desejado (BARRETO, 2005).

Para Wohlin (2004), embora se esteja desenvolvendo softwares mais complexos, os problemas básicos continuam os mesmos. Ainda, afirma que o principal fator envolvido no problema está relacionado às pessoas que desenvolvem o software.

De acordo com Dong *et al.* (2008), os recursos humanos geralmente representam o maior custo em projetos de software. Atividades como a alocação e agendamento de pessoas são realmente importantes e difíceis de serem executadas.

Segundo Laplante (2003), o elemento humano é um dos aspectos mais importantes, mas frequentemente negligenciado em gestão de projetos. Além disso, o autor afirma que o sucesso de um projeto está diretamente relacionado à qualidade da equipe participante e à maneira em que a gerência distribui as pessoas em um projeto.

Conforme Wang *et al.* (2006), o processo de desenvolvimento de software é altamente dependente da capacidade individual de engenheiros de software e de equipes de desenvolvimento. Entretanto, os recursos envolvidos no processo de desenvolvimento, especialmente as habilidades de recursos humanos que principalmente impactam neste processo, não têm sido bem considerados (WANG, XIAO, *et al.*, 2006). Além disso, o custo de um dado projeto está intimamente ligado ao esforço necessário para executá-lo, ou seja, à equipe de projeto (ANTONIOL, CIMITILE, *et al.*, 2004).

Uma vez que as pessoas desempenham um papel tão fundamental nos projetos de software, alocar pessoas, ou seja, determinar quem irá realizar cada tarefa em um projeto de software é uma atividade de grande importância (BARRETO, 2005). Conforme Acuña *et al.* (2006), este processo não é apenas crucial para gerar times produtivos, ele também pode ajudar organizações de software a desenvolver competências sistemáticas de longo prazo.

Para determinar quem é mais adequado a uma determinada atividade, os gerentes de projetos precisam considerar vários fatores, tais como, conhecimentos, habilidades técnicas e interpessoais, experiências, disponibilidade, entre outros (BARRETO, 2005). Entretanto, muitos gestores tipicamente executam esse processo baseados em sua experiência, conhecimento, percepção subjetiva e instinto. Embora esses meios possam ser úteis na alocação de pessoas, é importante ser extremamente cuidadoso ao realizar esse procedimento, já que envolve sentimentos de expectativas e satisfação de pessoas (ACUÑA, JURISTO e MORENO, 2006).

Segundo Miles e Klein (1998), a atribuição de tarefas pode ser uma experiência positiva ou negativa para um indivíduo. De acordo com Caudron (1997), fatores relacionados com a alocação de tarefas, tais como, identificar os anseios dos colaboradores, buscar o trabalho mais atrativo a um indivíduo, facilitar o desenvolvimento de um membro de equipe, influenciam na motivação das pessoas e estão diretamente ligados ao desempenho do trabalho a ser executado.

A alocação de pessoas a tarefas é um dos aspectos mais essenciais na área de gestão de projetos, já que uma designação inadequada pode ocasionar uma significativa perda para a empresa (ZHOU, 2008). O modelo proposto nessa dissertação busca facilitar a aprendizagem dos membros de um projeto, levando em consideração seus anseios. Essa abordagem visa trazer benefícios na execução de projetos, tais como, aumento da motivação dos colaboradores, aumento da aprendizagem organizacional, redução da rotatividade e melhoria no desempenho do trabalho.

1.2 Contribuições da Pesquisa

Essa dissertação apresenta como contribuição uma proposta para auxiliar nas decisões gerenciais em relação ao processo de alocação de tarefas aos participantes de uma equipe de projeto, visando obter maior desempenho na realização dos projetos de software e a melhoria de qualificação dos membros participantes.

A objetivo deste trabalho é a geração de um modelo de alocação que considere a aprendizagem das pessoas na execução de projetos. Este modelo busca o balanceamento entre os objetivos individuais (dos colaboradores) e organizacionais (custo, prazo, entre outros).

A partir deste modelo, busca-se responder as seguintes questões:

- É possível criar um sistema multiagente que auxilia a atribuição de tarefas a pessoas, levando em consideração os desejos profissionais de cada um?
- Quanto custa a um projeto levar em consideração os desejos profissionais dos colaboradores?
- Existe um estado de balanceamento onde seja satisfatório a empresa permitir a aprendizagem dos membros (em temas de interesse dos mesmos), mesmo obtendo maior duração e/ou maior custo no projeto?

A partir dessas questões, chegou-se à hipótese de pesquisa básica desse trabalho: *“Considerar a motivação e o desenvolvimento das pessoas ao alocar tarefas pode trazer benefícios para a empresa”*. Essa hipótese se relaciona com as questões propostas, pois se pretendeu criar um modelo que considere os desejos profissionais dos membros participantes de um projeto, mensurar o custo de levar

em consideração esses desejos e, buscar um estado de balanceamento entre os objetivos individuais e organizacionais, conforme resultados apresentados.

Outras contribuições do trabalho incluem:

- a) Elaborar um modelo de simulação para alocação de recursos humanos em projetos de desenvolvimento de software;
- b) Configurar um modelo computacional para execução da simulação proposta.

1.3 Justificativa

A alocação de pessoas a tarefas é um problema complexo. Como há uma grande quantidade diferente de combinações de alocações possíveis, é difícil para um gestor considerar todas as possibilidades envolvidas, visualizar o impacto de cada uma delas sobre algum fator, como custo, qualidade, entre outros, e escolher a melhor equipe para um dado projeto (BARRETO, 2005).

Dessa forma, a complexidade e a natureza da gestão de projetos tornam o uso de técnicas como a simulação algo muito atrativo aos gerentes, já que se pode simular as características dos agentes envolvidos, os efeitos de eventos externos, a incerteza sobre a duração das tarefas, entre outros (JOSLIN e POOLE, 2005).

Wickenberg e Davidsson (2002) crêem que a simulação é uma maneira relativamente eficiente para permitir que gestores possam elaborar várias configurações diferentes de processos, compreender melhor os efeitos de várias políticas, experimentando mudanças no processo real com custos em tempos e recursos relativamente pequenos (ALVES, 2009).

Segundo Jinghua e Wenjian (2005), a motivação para incorporar simulações no planejamento de projetos é bastante clara. As simulações oferecem possibilidades de representar a complexidade necessária para raciocinar de forma realista sobre o projeto. Ainda, as simulações oferecem o potencial para representar a complexidade de um projeto em desenvolvimento, as incertezas inerentes, as características individuais dos agentes envolvidos e como eles podem afetar o estado do projeto

Dentro do contexto de simulações, o uso de sistemas multiagentes é bastante adequado para modelar sistemas heterogêneos, autônomos e com atores pró-ativos, tais como sistemas centrados em seres humanos (SIEBERS e AICKELIN, 2007).

Conforme Jennings e Wooldridge (1998), muitos ambientes (incluindo a maioria das organizações) são naturalmente modelados como sociedades de agentes. Bonabeau (2002) afirma que o uso de multiagentes, em muitos casos, é a forma mais natural para descrever e simular um sistema composto por entidades comportamentais.

Dessa forma, optou-se pelo uso de simulações multiagentes nesse trabalho, buscando auxiliar gerentes de projetos em decisões de planejamento, melhorando suas previsões e diminuindo os riscos na execução do projeto.

1.4 Estrutura da Dissertação

O texto está organizado como segue. No capítulo 2, aborda-se conceitos de gestão de projetos de software e recursos humanos, bem como os aspectos que devem ser considerados ao alocar pessoas à atividades nesses projetos. No capítulo 3, aborda-se os conceitos envolvidos em sistemas multiagentes, suas características e classificações. No capítulo 4, aborda-se os métodos utilizados para desenvolver este trabalho e as etapas de seu desenvolvimento. No capítulo 5, apresenta-se o modelo construído, suas premissas, características e plataforma utilizada. No capítulo 6, são apresentados os resultados encontrados. Por fim, no capítulo 7 são apresentadas as considerações finais, limitações do estudo, trabalhos correlatos e trabalhos futuros.

2 ALOCAÇÃO DE PESSOAS EM PROJETOS DE SOFTWARE

Este capítulo apresenta uma revisão bibliográfica sobre gestão de projetos de software, bem como, sobre o processo de alocação de tarefas e recursos em projetos e os fatores que o influenciam.

2.1 Conceitos Relacionados a Projetos

Segundo o PMI (2008, p. 11), “um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo”. A partir dessa definição podem-se salientar as seguintes características:

- Natureza temporária: indica um início e término definido.
- Produtos e serviços exclusivos: mesmo que haja elementos repetitivos em entregas do projeto, há circunstâncias, projetos, prazos, pessoas envolvidas diferentes.

Para Kerzner (2001), um projeto é qualquer série de atividades e tarefas que:

- Têm um objetivo específico a ser concluído dentro de determinadas especificações;
- Têm início e término definido;
- Têm limites de financiamento (se aplicável);
- Consomem recursos humanos e não-humanos (ou seja, o dinheiro, as pessoas, equipamentos);
- Sejam multifuncionais (isto é, abarcam várias linhas funcionais).

Em relação ao conceito de gestão de projeto, há vários que podem ser citados. O PMI (2008, p. 12) descreve gerenciamento de projetos como “a aplicação do conhecimento, habilidades, ferramentas e técnicas às atividades de do projeto a fim de atender aos seus requisitos”. Além disso, gerenciar um projeto inclui a identificação de requisitos, adaptação às diferentes necessidades dos *stakeholders*, balanceamento das restrições conflitantes de escopo, qualidade, cronograma, orçamento, recursos e riscos.

Para Weng *et al.* (2009), a gerência de projetos é uma atividade de gestão para analisar e gerenciar custos, pessoas, cronograma, qualidade e riscos de modo a garantir que um projeto seja realizado com êxito, de acordo com um custo, cronograma e qualidade previamente acordados durante todo o ciclo de vida do

projeto. Ainda, conforme Reiss (1999), um projeto é uma atividade humana que permite atingir objetivos claros contra uma escala de tempo.

Mesmo que haja vários conceitos diferentes de gestão de projetos na literatura, as palavras custo, tempo e qualidade sempre estão presentes (ATKINSON, 1999).

Estas variáveis são descritas através de um triângulo denominado “Triângulo das Restrições”, apresentado na Figura 2.1. Esse triângulo contempla três variáveis (tempo, custo e escopo) que necessitam ser balanceadas para alcançar o sucesso do projeto (PHILIPS, 2010, p. 424). O conceito por trás desse triângulo diz que, uma vez estabelecido um equilíbrio entre estas três dimensões, qualquer mudança em uma delas obrigatoriamente afetará uma das duas outras, a não ser que se altere a qualidade do projeto (CARVALHO, 2004, p. 50).

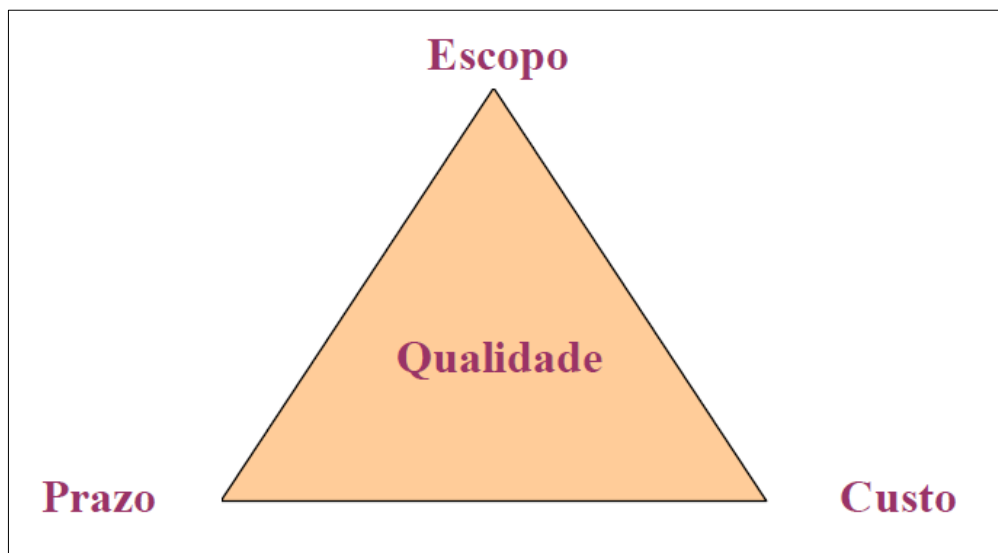


Figura 2.1 – Triângulo das Restrições
Fonte: (CARVALHO, 2004, p. 50)

Ainda dentro de conceitos relativos à gerência de projetos, é relevante a este trabalho definir os termos programa e portfólio. Segundo o PMI (2008, p. 14), “um portfólio refere-se a um conjunto de projetos ou programas e outros trabalhos, agrupados para facilitar o gerenciamento eficaz desse trabalho a fim de atingir os objetivos de negócios estratégicos”.

Já um programa, conforme o PMI (2008, p. 15), “é definido como um grupo de projetos relacionados gerenciados de modo coordenado para a obtenção de benefícios e controle que não estariam disponíveis se eles fossem gerenciados

individualmente”. Na Tabela 2.1 e na Figura 2.2 é possível visualizar as diferenças entre programa e portfólio de projetos.

Projeto	Programa	Portfólio
Escopo reduzido, entregas específicas	Escopo abrangente, pode sofrer modificações para atender expectativa de negócios	Possui escopo de negócios que acompanha a estratégia da organização
Gerente de projetos tenta minimizar mudanças	Gerente de programa espera e anseia por mudanças	Gerente de portfólio monitora continuamente as mudanças no ambiente
Sucesso mensurado por tempo, escopo e custo	Sucesso mensurado por ROI, novas capacidades e entrega de benefícios	Sucesso é mensurado em termos de desempenho agregado dos componentes
Estilo de liderança focado em realização de atividades	Liderança focada em gerenciamento de relacionamentos, conflitos, aspectos políticos	Estilo de liderança focado em adição de valor às decisões do portfólio
Gerente de projetos gerencia técnicos, especialistas	Gerentes de programa gerenciam gerentes de projetos	Gerentes de portfólio podem gerenciar a equipe de gerenciamento de portfólio
Gerente de projetos faz parte da equipe e motiva através das suas habilidades	Gerente de programa é um líder e provê visão e liderança	Gerentes de portfólio são líderes que provêm compreensão e intuição
Gerente de projetos com foco em planejamento detalhado, com objetivo nas entregas	Gerente de programa cria planos de alto nível que guiam os projetos (que detalham)	Gerente de portfólio cria e mantém processos e comunicação no portfólio
Gerente de projetos monitora e controla atividades e os produtos gerados	Gerente de programa monitora projetos e outras iniciativas através de estruturas de governança	Gerentes de portfólio monitoram o desempenho agregado e indicadores de valor

Tabela 2.1 – Comparação entre Projeto, Programa e Portfólio
Fonte: Vargas (2011)

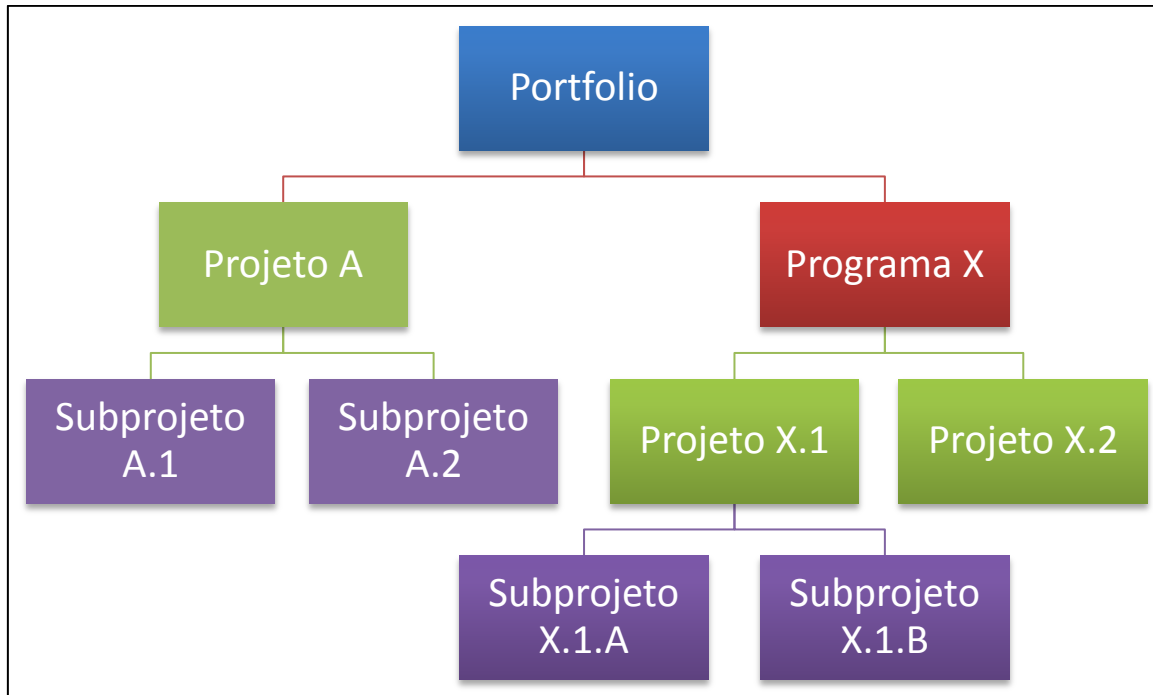


Figura 2.2 – Diagrama ilustrativo de Projetos, Programas e Portfolios
 Fonte: Adaptado de PMI (2008, p. 14)

É importante esclarecer que, quando se usa o termo multi-projeto, refere-se ao gerenciamento de programas de projeto, como utilizado por Dong *et al.* (2008) em seu trabalho.

2.2 Gerência de Projetos de Software

Gerenciar projetos de software não é o mesmo que gerenciar outros projetos. Conforme Sommerville (2007, p. 62), há distinções na engenharia de software em relação à outras engenharias que tornam a gestão desses projetos particularmente difícil, tais como:

- Intangibilidade do produto: dificuldade de avaliar o progresso do projeto, já que o produto não é visível.
- Inexistência de processos-padrões de software: os processos de software variam drasticamente de uma organização para outra e, mesmo com toda a evolução nesta área, ainda há uma dificuldade de prever problemas de desenvolvimento.
- Unicidade de projetos de software de grande porte: esse tipo de projeto geralmente é diferente dos anteriores e, mesmo com experiência da equipe de projeto, é difícil prever problemas.

Hughes e Cotterell (2006) também mencionam características específicas que tornam os projetos de desenvolvimento de software diferentes dos demais. São elas:

- Invisibilidade: o processo de desenvolver software não é visual como outros projetos, sendo, portanto, difícil monitorar, estimar e avaliar seu progresso.
- Complexidade: além do processo de desenvolvimento existem as fases de implantação e manutenção que podem ser distribuídas e possuir interfaces com muitos sistemas existentes (NIENABER, 2008).
- Conformidade: geralmente projetos tradicionais envolvem recursos físicos imutáveis, enquanto projetos de software envolvem uma variedade de recursos baseados em requisitos de clientes, que podem ser inconsistentes por causa de problemas organizacionais, tais como, memória coletiva, comunicação interna ou tomada de decisões efetivas.
- Flexibilidade: um dos pontos fortes de softwares é a facilidade de alterações para se adequar aos padrões das organizações, o que ocasiona o desenvolvimento de sistemas que sejam flexíveis a tais modificações, possuindo alto nível de mudanças.

Segundo Nienaber (2008, p. 25), “a natureza única da gestão de projetos de software contribui para as dificuldades geralmente encontradas e para o fracasso desse tipo de projeto”. Assim, as organizações devem buscar alternativas para gerenciar projetos de software, visando reduzir falhas e melhorar a qualidade de seus produtos e serviços (CISCON, 2009).

2.3 Processos de Gestão de Projetos

Os projetos variam em tamanho e complexidade e podem ser mapeados segundo a ilustração do ciclo de vida da Figura 2.3. Os ciclos de vida do projeto geralmente definem que trabalho deve ser realizado em cada etapa, as entregas geradas e as pessoas envolvidas, permitindo a facilitação de controle de andamento do projeto.

Segundo o PMI (2008), a maioria dos ciclos de vida de projetos compartilha diversas características comuns, tais como:

- Fases geralmente sequenciais;

- Níveis de custos de pessoal baixos inicialmente, depois atingem os valores máximos durante as fases intermediárias e caem rapidamente, quando o projeto é finalizado;
- Influência das partes interessadas, riscos e incertezas são maiores durante o início do projeto;
- Capacidade de influenciar as características finais do produto do projeto, sem impacto significativo sobre os custos, é mais alta no início e torna-se cada vez menor conforme o projeto progride para o seu término.

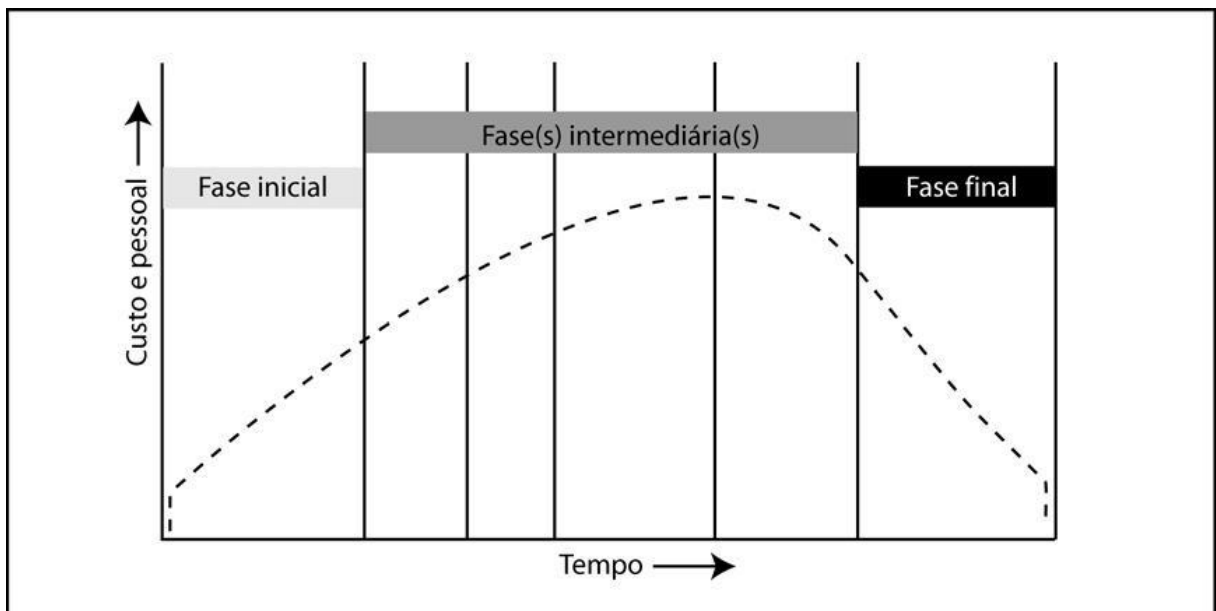


Figura 2.3 – Ciclo de vida do projeto
Fonte: Adaptado de PMI (2008, p. 22)

Dentro do andamento do projeto, há certas atividades a serem executadas, que podem variar de acordo com o método escolhido. Todavia, essas atividades geralmente envolvem definir o que deve ser feito, decidir como fazê-lo, controlar sua realização e avaliar o que já foi desenvolvido (CHANG, CHAO, *et al.*, 1998).

A questão de definir o que está sendo feito normalmente assume a forma de um plano. O plano do projeto estabelece os recursos disponíveis para o projeto, as entregas do projeto e suas atividades, bem como um cronograma para realizar o trabalho (SOMMERVILLE, 2007).

A questão do "como" refere-se à alocação de recursos, tais como, cronograma, orçamento, pessoas (CHANG, CHAO, *et al.*, 1998). Com respeito à

alocação de pessoas, considera-se esta fase extremamente importante, pois influencia diretamente no sucesso do projeto (BARRETO, 2005).

Em relação ao controle e avaliação, para Hashim e Keshlaf (2009), são atividades importantes pois, através do monitoramento e controle, pode-se verificar se um projeto está em conformidade com a programação prevista e resolver desvios das atividades programadas. Já Ming e Yang (2009) mencionam a relevância da avaliação de projetos de TI, que é uma atividade presente em todas as fases de um projeto, incluindo o período de preparação, construção e operação. A avaliação influencia no processo de tomada de decisão desses projetos, reduzindo o risco, aumentando a experiência de seus participantes e sua taxa de sucesso.

Segundo o PMI (2008), a gestão de projetos possui nove áreas de conhecimento que são: gestão de integração, escopo, tempo, custo, qualidade, comunicação, riscos, aquisições e recursos humanos. Além disso, Schwalbe (2008) divide essas áreas de conhecimento em dois tipos: áreas de conhecimento principal e áreas de conhecimento facilitadoras. As áreas de conhecimento principal conduzem a objetos específicos do projeto. São elas: escopo, tempo, custo e qualidade. As demais áreas são consideradas facilitadoras, pois são os processos nos quais os objetivos de projetos podem ser atingidos. Engloba as áreas de recursos humanos, comunicação, risco e aquisição. A nona área, de integração, é uma função geral que afeta e é afetada por todas as outras áreas de conhecimento. Ela envolve a coordenação das outras áreas, buscando antecipar e lidar com questões do projeto, bem como, tomando decisões sobre o que é melhor para o projeto como um todo.

Ainda, para o PMI (2008), essas áreas de conhecimento são responsáveis pelos seguintes processos:

- Gerenciamento da integração: inclui os processos e as atividades necessárias para identificar, definir, combinar, unificar e coordenar os vários processos e atividades dos grupos de processos de gerenciamento. A ideia principal da integração é integrar e coordenar as outras áreas de conhecimento.
- Gerenciamento do escopo: inclui os processos necessários para assegurar que o projeto inclui todo o trabalho necessário, e apenas o necessário, para terminar o projeto com sucesso. Esta área define e controla o que está e o que não está incluso no projeto.

- Gerenciamento do tempo: inclui os processos necessários para gerenciar o término pontual do projeto, como definir e sequenciar atividades, estimar os recursos e durações das atividades, desenvolver e controlar o cronograma.
- Gerenciamento de custos: inclui os processos envolvidos em estimativas, orçamentos e controle dos custos, de modo que o projeto possa ser terminado dentro do orçamento aprovado.
- Gerenciamento da qualidade: inclui os processos e as atividades da organização executora que determinam as políticas de qualidade, os objetivos e as responsabilidades, de modo que o projeto satisfaça às necessidades para as quais foi empreendido.
- Gerenciamento das comunicações: inclui os processos necessários para assegurar que as informações sejam geradas, coletadas, distribuídas, armazenadas, recuperadas e organizadas de maneira oportuna e apropriada.
- Gerenciamento de riscos: inclui os processos de planejamento, identificação, análise, planejamento de respostas, monitoramento e controle de riscos de um projeto. Os objetivos do gerenciamento de riscos são aumentar a probabilidade e impacto dos eventos positivos e reduzir a probabilidade e o impacto dos eventos negativos no projeto.
- Gerenciamento de aquisições: inclui os processos necessários para comprar ou adquirir produtos, serviços ou resultados externos à equipe do projeto.

A gestão de recursos humanos do projeto será discutida na próxima seção com mais detalhes, visto que é tema deste trabalho.

2.4 Gestão de Pessoas em Projetos

Para o PMI (2008, p. 181), “o gerenciamento dos recursos humanos do projeto inclui os processos que organizam e gerenciam a equipe do projeto, ou seja, as pessoas com papéis e responsabilidades designadas para a conclusão do projeto”. Os principais processos envolvidos são:

- Desenvolvimento do plano de recursos humanos: identificação das funções, responsabilidades e habilidades necessárias.

- Mobilização da equipe de projeto: designação das pessoas adequadas às tarefas existentes.
- Desenvolvimento da equipe de projeto: melhoria de competências, interação e ambiente global da equipe.
- Gerenciamento da equipe de projeto.

Segundo a norma NBR ISO 1006 (2000, p. 10), “os processos relacionados às pessoas visam criar um ambiente no qual o pessoal possa contribuir efetiva e eficientemente para o projeto. São eles:

- Definição da estrutura organizacional do projeto: inclui a identificação de autoridade e responsabilidades.
- Alocação da equipe: selecionar e nomear pessoal suficiente, com competência apropriada para corresponder às necessidades do projeto.
- Desenvolvimento da equipe: desenvolver habilidades individuais e coletivas para melhorar o desempenho do projeto.

Ainda, Schnaider (2003), o processo de gerência de recursos humanos compreende as seguintes etapas:

- Identificação das competências necessárias dos recursos humanos alocados no projeto.
- Seleção de recursos humanos adequados.
- Monitoramento da alocação de recursos humanos: acompanhar o andamento das atividades, atualizando o plano de alocação de recursos humanos a cada alocação e desalocação.
- Avaliação do desempenho dos recursos humanos.

Combinando os processos dos autores anteriormente citados, será utilizado o fluxo da Figura 2.4 para gerenciar pessoas em projetos.

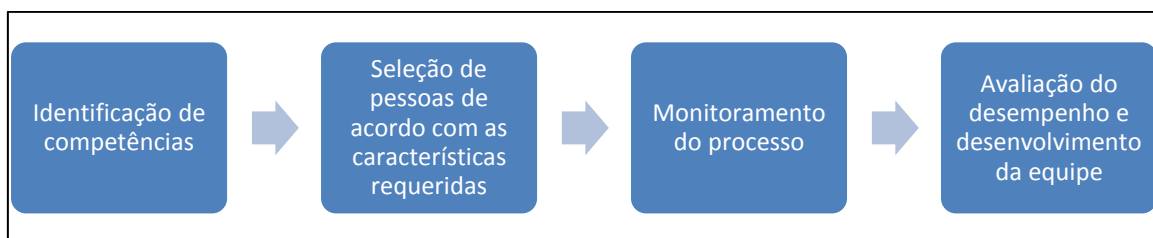


Figura 2.4 – Processo de gestão de pessoas em projetos.

Fonte: Adaptada pela autora dos autores PMI (2008), Schnaider (2003) e NBR ISO 1006 (2000).

2.4.1 Identificação de Competências

O termo competência possui vários conceitos na literatura. Para Dutra (2004, p. 29), “competência é o conjunto de qualificações que permite à pessoa um desempenho superior em seu trabalho ou situação”. Ainda, o autor afirma que o conceito de competência está em construção e tem se mostrado muito efetivo para explicar a realidade vivida pelas empresas na gestão de pessoas.

Fleury (2002, p. 55) define competências como “um saber agir responsável e reconhecido, que implica em mobilizar, integrar, transferir conhecimentos, recursos, habilidades que agregam valor econômico à organização e valor social ao indivíduo”. Para o autor as competências humanas podem ser entendidas como um conjunto de conhecimentos, habilidades e atitudes que resulta em uma entrega ou aplicação prática.

De acordo com Barbazette (2005), um conjunto de competências é conhecido como modelo de competências e é uma coleção de comportamentos suportados por conhecimento, habilidades e atitudes que se relacionam a um papel específico. Esse modelo (ver figura 2.5) é circular, por causa de uma variedade de conhecimentos, habilidades e atitudes de apoio ao par comportamento-desempenho que geram um resultado final.

Para Goulart (2006), a utilização do conceito de competência exige o esclarecimento dos termos conhecimentos, habilidades e atitudes, já que compõem este termo. Segundo Wood Jr. e Picarelli Filho (2004, p. 106), “os conhecimentos constituem um conjunto de conceitos, técnicas, teorias e metodologias relacionadas ao trabalho. É o saber adquirido, que pode ser transmitido e aprendido”.

Já a expressão habilidade, conforme Goulart (2006, p. 90), “refere-se à capacidade do indivíduo de aplicar seus conhecimentos na prática, é o saber fazer. É constituída por aptidões aprendidas pelo sujeito, que são necessárias à atuação em papéis ocupacionais determinados.” Existem habilidades físicas, intelectuais e sociais.

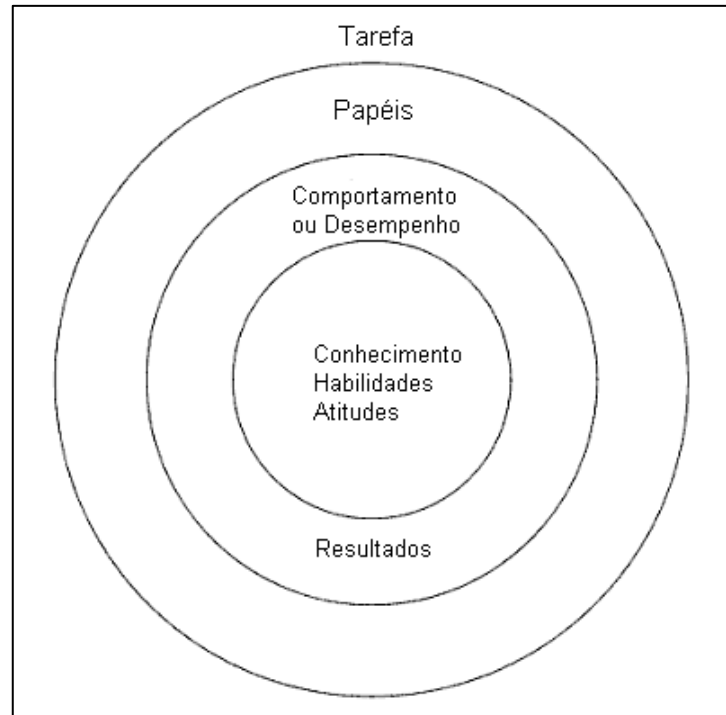


Figura 2.5 – Modelo de competência.
 Fonte: Adaptado de Barbazette (2005, p. 15).

Para Lustri, Miura e Takahachi (2005), as atitudes referem-se ao saber ser. Dizem respeito ao que se convencionaram chamar de competências comportamentais, tais como, habilidades de relacionamento pessoal, flexibilidade, pró-atividade, entre outras.

Neste trabalho será utilizado o conceito de competência Barbazette (2005), por ser um dos conceitos mais difundidos nessa área. O primeiro passo no processo de gestão de pessoas em projetos é identificar essas competências. Um método comumente utilizado para identificar competências é o etnográfico, que consiste em uma combinação de várias técnicas, como observação em campo, entrevistas e questionários (RIVERA-IBARRA, RODRIGUEZ-JACOBO e SERRANO-VARGAS, 2010). Ainda, pode-se utilizar reuniões, *brainstormings*, análise de lições aprendidas, entre outros (SCHNAIDER, 2003). Além disso, para facilitar a identificação das competências necessárias, Dingsoyr e Royrvik (2001) sugerem que elas sejam agrupadas em categorias e que cada competência seja graduada através da atribuição de um nível. Por exemplo, na Tabela 2.2, pode-se visualizar uma determinada pessoa que tem uma competência técnica de projeto de bancos de dados, na tecnologia MySQL com nível de experiência X (que poderia ser medido em anos, em níveis de conhecimento, entre outros).

Disciplina	Engenharia de Software
Área	Projeto
Categoria	Banco de Dados
Tecnologia	MySQL
Nível	3

Tabela 2.2 – Exemplo de uma classificação de competência

O produto gerado durante a identificação de competências é a lista de competências do projeto, que contém todas as competências identificadas como necessárias à realização de cada tarefa do projeto (SCHNAIDER, 2003). O modelo de competências utilizado neste trabalho foi adaptado do framework sugerido pelos autores Rivera-Ibarra, Rodriguez-Jacobo e Serrano-Vargas (2010) e será apresentado com detalhes na seção 5.5.2.

2.4.2. Seleção de Pessoas

Após identificadas as competências necessárias para as tarefas do projeto, o gerente necessita buscar os profissionais mais adequados para executá-las. Duas pessoas com o mesmo cargo podem diferir em diversos aspectos, tais como (PFLEEGER, 2004):

- Capacidade para desempenhar o trabalho;
- Interesse no trabalho;
- Experiência com aplicações semelhantes;
- Experiência com ferramentas ou linguagens semelhantes;
- Experiência com técnicas semelhantes;
- Experiência com ambiente de desenvolvimento semelhante;
- Treinamento;
- Capacidade para se comunicar com outras pessoas;
- Capacidade para compartilhar responsabilidades com outras pessoas;
- Habilidades de gerenciamento.

Cada uma dessas características pode afetar a capacidade de um indivíduo para realizar seu trabalho (BARRETO, 2005). Portanto, selecionar pessoas para as

atividades do projeto não deve levar em conta apenas as competências requeridas. É importante considerar também as preferências dos colaboradores envolvidos, pois algumas tarefas podem inibir sua produtividade (IVANCEVICH, 1979).

O processo de seleção baseia-se em dados e informações das análises e especificações da tarefa a ser executada em comparação com o perfil de características da equipe de projeto. Quando as especificações das tarefas são maiores que as características do colaborador, esta pessoa não está apta a executar essa tarefa. Quando as variáveis são iguais, o membro do projeto reúne as condições ideais para a execução da atividade. Entretanto, se as características da pessoa são superiores às especificações da tarefa, dada uma faixa de aceitação, o colaborador é superdotado para esta atividade (CHIAVENATO, 2004).

Nesta fase é gerado o plano de alocação de recursos humanos, documento que inclui a lista de todos os membros da equipe participantes alocados nas tarefas durante o ciclo de vida do projeto (SCHNAIDER, 2003).

2.4.3. Monitoramento do Processo

Nesta etapa busca-se monitorar o andamento da alocação e desalocação dos recursos humanos, verificando se este processo está ocorrendo nas datas programadas com os resultados esperados. Caso não ocorra, é papel do gerente de projetos negociar os prazos e requisitos, adequando o plano de alocação de acordo com novas situações.

Há muito tempo projetos de software sofrem de vários tipos de problemas, como cronograma e orçamento acima do planejado, qualidade baixa do produto, entre outros (LIU, WANG e XIAO, 2009). Nesta área geralmente ocorrem mudanças durante o ciclo de vida do projeto. Normalmente é impossível definir o problema, projetar uma solução e implementar o software sem que haja mudanças. Especialmente, para softwares complexos, se torna difícil entender todos os requisitos no início do projeto (BHATTI, HAYAT, *et al.*, 2010).

O motivo das mudanças pode ser proveniente de fatores internos ou externos. Fatores externos podem ser provenientes de pressão de clientes, mudança das necessidades de negócio das empresas dos clientes. Já fatores internos podem se referir ao entendimento superficial ou errôneo das necessidades, inviabilidade técnica dos requisitos do cliente, entre outros (WANG e LAI, 2001).

O contínuo processo de mudanças de requisitos pode afetar o custo, cronograma e qualidade do projeto de software e incapacidade de lidar com essas mudanças pode levar a inconsistências nos requisitos do sistema e resultar em falhas do projeto (BHATTI, HAYAT, *et al.*, 2010).

As técnicas tipicamente utilizadas no monitoramento da alocação de recursos humanos são fundamentadas nos métodos de gerenciamento de riscos de um projeto e se baseiam no acompanhamento do plano de alocação de recursos humanos (SCHNAIDER, 2003). Neste passo o gerente de projeto compara a alocação de pessoas real com a planejada, verificando os profissionais participantes e as datas em que efetivamente ocorreram as alocações. É interessante documentar os problemas enfrentados durante o ciclo de vida do projeto, para evitar erros futuros.

2.4.4. Avaliação do Desempenho e Desenvolvimento da Equipe

A questão mais importante para a gestão de recursos humanos é encontrar profissionais de alta qualidade para as empresas (JING, 2009). Na era do conhecimento, todas as organizações gradualmente aceitam a ideia de "capital humano" e concordam que as pessoas são o mais importante recurso econômico que uma empresa pode ter (WANG e JIANG, 2010).

Assim como selecionar a pessoa adequada para executar uma dada tarefa, avaliar seu desempenho é algo crucial para as organizações (JING, 2009). A avaliação do desempenho não é apenas a avaliação de resultados, mas a busca de melhoria contínua para as organizações. Através de sistemas de avaliação, as companhias podem desenvolver sistemas de reconhecimento e premiação para ajudar a manter a motivação e o espírito de equipe, contribuindo assim para a qualidade dos serviços da empresa (XIE e TANG, 2008).

Para Durai (2010), a avaliação de desempenho é um passo vital no processo de gestão de recursos humanos e tem como principais objetivos:

- Identificar as lacunas de desempenho: a avaliação de desempenho ajuda a determinar a lacuna existente entre o desempenho do profissional e o desempenho esperado ou desejado pela organização.
- Prover informações para promoção, transferência ou demissão;
- Ajudar a determinar salários ou recompensas;

- Ajudar a melhorar a efetividade dos colaboradores: os resultados da avaliação de desempenho podem ser usados para decidir se um colaborador ou uma equipe necessita treinamento e desenvolvimento adicional, focando a capacitação em fortalecer suas fortalezas e diminuir suas deficiências.
- Melhorar relações interpessoais: pode ser usada como um mecanismo de comunicação entre superiores e subordinados.
- Identificar insatisfação dos colaboradores.

Ainda, Durai (2010) afirma que a avaliação de desempenho é uma boa opção para determinar habilidades de um profissional para cumprir suas tarefas. Um sistema de avaliação de desempenho efetivo deve ter foco não apenas nos objetivos organizacionais, mas também no colaborador, buscando ajudá-lo a melhorar seu desempenho e desenvolver seu plano de carreira.

Com a informação da avaliação de desempenho, a empresa pode dedicar-se a auxiliar seus colaboradores a se desenvolver. Segundo Chiavenato (2010), existem vários métodos de desenvolvimentos de pessoas, tais como, rotação de cargos dentro da empresa, posições de assessoria, atribuição de comissões, participação em cursos e seminários, exercícios de simulação, estudos de casos, jogos de empresas, aprendizagem prática, entre outros. Um dos mais comuns, a aprendizagem prática, é uma técnica de treinamento onde o colaborador se dedica a analisar e resolver problemas em certos projetos ou em outros departamentos.

Para avaliar o desempenho de recursos humanos em projetos de software, Klein *et al.* (2001) propõe sete grupos de medidas. São elas:

- Medidas de Qualidade: avaliam a qualidade do trabalho realizado como um todo, podendo incluir a adesão a normas e padrões estabelecidos, a utilização adequada de ferramentas e a busca da eficiência.
- Medidas de Gerência: avaliam o nível de comprometimento com os objetivos do projeto de uma forma geral, podendo incluir o comprometimento com cronogramas e custos, e o atendimento às especificações de requisitos definidas para o sistema.
- Medidas de Tarefas Genéricas: avaliam a competência na realização de tarefas genéricas, podendo incluir a capacidade de antecipar problemas do cliente e habilidades de comunicação e persistência.

- Medidas Pessoais: avaliam a atitude “orientada ao cliente”, podendo incluir habilidades em negociação, gerência de mudanças e entendimento das necessidades do cliente.
- Medidas de Confiabilidade: avaliam a confiança que o cliente tem na equipe do projeto, podendo incluir o grau de comportamento ético empregado na solução de problemas emergenciais.
- Medidas de Liderança e Trabalho em Equipe: avaliam o grau de liderança e de comprometimento com as boas práticas do trabalho em equipe.
- Medidas de Sucesso: avaliam a capacidade de executar múltiplas tarefas concorrentemente, de gerenciar interrupções e novas atribuições.

Segundo Schnaider (2003), a avaliação da equipe de projeto deve ser realizada periodicamente ao longo do ciclo de vida do projeto de software, e uma vez após a conclusão deste. A avaliação periódica permite que eventuais maus desempenhos possam ser logo identificados e ações corretivas adotadas brevemente. Já a avaliação após a conclusão do projeto é importante para gestão do conhecimento, analisando os erros e acertos na execução do projeto, visando melhorias nos projetos futuros.

2.5 Objetivos Individuais e Organizacionais

Os objetivos individuais de um colaborador estão intimamente ligados com sua motivação na execução de atividades. Conforme Hunt e Osborn (1999, p. 86), “motivação se refere às forças dentro de uma pessoa responsáveis pelo nível, direção e persistência do esforço dispendido no trabalho”. Para os autores, nível significa a quantidade de esforço que a pessoa emprega. Já a direção é o que a pessoa opta por fazer quando possui muitas alternativas sobre o que realizar. Por último, a persistência se refere a quanto tempo a pessoa continua numa determinada ação.

Existem várias teorias na literatura sobre motivação humana, tais como, Teoria da Hierarquia das Necessidades Humanas (MASLOW, 1954), Teoria dos Dois Fatores (HERZBERG, 1966), Teoria X e Y (MCGREGOR, 1985), entre outros. Segundo Fleury (2002), a motivação é um processo interno, não é possível motivar os indivíduos, pois cada necessidade é individual e singular. De qualquer forma, a

empresa pode trabalhar os aspectos motivacionais, ajudando os seus colaboradores a se motivarem.

Segundo um estudo realizado por Sirota *et al.* (2005), os aspectos motivacionais mais comuns no trabalho são: igualdade (salário justo, benefícios e segurança no emprego); realização (orgulho de seu trabalho e de suas realizações e, por extensão, elogio, reconhecimento e crescimento no emprego); relações cordiais com os colegas. É importante ressaltar o item realização, afinal se os funcionários se sentem bem com sua atividade e visualizam oportunidades de crescimento, conseqüentemente se sentirão mais motivados para trabalhar diariamente e se esforçarão mais para realizar seu trabalho (SILVERSTEIN, 2009).

Dentre desse contexto, é importante diferenciar os objetivos individuais dos organizacionais. Os objetivos individuais são variáveis de acordo com a percepção de cada pessoa. Já os objetivos organizacionais resultam da vontade grupal da qual o indivíduo faz parte, independente de sua expectativa motivacional (COLARES, 2008).

Como a tendência de uma organização bem sucedida é crescer, muitas vezes a empresa necessita reduzir custos, prazos e outros fatores. Os conflitos organizacionais ocorrem, quando, por exemplo: a redução de custos esbarra na expectativa de melhores salários; o aumento da lucratividade conflita com maiores benefícios sociais. Assim, a empresa necessita estar continuamente pensando em um processo de integração de objetivos, visando evitar desmotivação e a redução da produtividade (COLARES, 2008). Ainda, para Argyris apud Chiavenato (1997, p. 575), “as organizações que apresentam alto grau de integração entre objetivos individuais e organizacionais são mais produtivas do que as demais”.

2.6 Considerações Finais

As pessoas são o fator menos formalizado nos modelos atuais de processo de software. Contudo, sua importância é evidente: elas se comportam de forma não-determinística e subjetiva, tendo um impacto decisivo sobre os resultados do desenvolvimento de software, que é uma atividade essencialmente intelectual e social (SOMMERVILLE e RODDEN, 1996).

Neste capítulo foi apresentada uma breve revisão bibliográfica sobre o processo de gestão de pessoas em projetos de desenvolvimento de software. Além

disso, foi mencionado o processo de gestão de projetos com suas áreas de conhecimento, bem como as diferenças que existem em desenvolvimento de software em relação a outros projetos.

Por fim, foram expostas as principais etapas do processo de alocação de pessoas, visando encontrar a pessoa mais adequada a uma dada tarefa, ressaltando a importância de se considerar os planos de carreira dos colaboradores, além dos objetivos estratégicos organizacionais.

3. AGENTES DE SOFTWARE

Este capítulo trata dos conceitos relacionados aos agentes inteligentes de software e seus sistemas, tais como, sua definição e características, classificações, arquiteturas e modelagem. Além disso, apresenta algumas das possíveis aplicações desses sistemas.

3.1 Definição de Agentes Inteligentes e suas Características

Há vários conceitos para agentes de software. Um deles, proposto por Wooldrige (2002, p. 15), define agente como “uma entidade de software que está situada em algum ambiente e é capaz de ação autônoma neste ambiente a fim de atingir seus objetivos de projeto”.

Segundo Bonabeau (2002), agentes são entidades que avaliam sua situação e tomam decisões a partir de um conjunto de regras. Esses agentes podem executar vários comportamentos apropriados para o sistema que eles representam. Além disso, são capazes de interagir com outros agentes e evoluir, permitindo novos comportamentos, não previstos anteriormente.

Para Macal e North (2005), agentes inteligentes possuem as seguintes propriedades:

- É identificável, com um conjunto de características e regras que regem os seus comportamentos e sua capacidade de tomada de decisão.
- É situado, vivendo em um ambiente no qual interage com outros ambientes, a partir de protocolos de interação e comunicação. Possui a capacidade de responder a esse ambiente.
- É dirigido a objetivos, têm metas a atingir em relação a seu comportamento.
- É autônomo e auto-dirigido, podendo operar independentemente do ambiente que está inserido.
- É flexível e têm a capacidade de aprender e se adaptar ao longo do tempo com base em sua experiência.

Jennings e Wooldrige (1998, p. 4) especificam que um agente inteligente deve ter as seguintes características:

- Reatividade: agentes devem perceber o ambiente em que estão inseridos e responder em um tempo hábil às mudanças que ocorrem nele;
- Pró-atividade: agentes não podem simplesmente agir em resposta ao ambiente em que estão inseridos, eles devem apresentar um comportamento oportunista, dirigido a objetivos e tomar a iniciativa quando for apropriado.
- Comportamento social: agentes devem ter a capacidade de interagir quando for necessário com outros agentes artificiais ou humanos, a fim de alcançar seus objetivos e/ou ajudar outros agentes com suas próprias atividades.

3.2 Sistemas Multiagentes (MAS) e suas Classificações

Um sistema multiagente é um sistema baseado na ideia na qual um ambiente de trabalho cooperativo incluindo componentes de softwares sinérgicos pode lidar com problemas que seriam difíceis de serem resolvidos usando uma abordagem de computação tradicional centralizada. Agentes de software com as capacidades de autonomia, reatividade, pró-atividade e habilidade social interagem de um modo dinâmico e flexível para resolver problemas de forma mais eficiente (CERVENKA e TRENCANSKY, 2000) da mesma forma que as pessoas fazem em suas vidas (WOOLDRIDGE, 2002).

De acordo com O'Hare e Jennings (1996), um sistema multiagente é um sistema de inteligência artificial composto por uma população de agentes autônomos que cooperam entre eles para alcançar objetivos comuns, enquanto simultaneamente perseguem objetivos individuais.

Ainda, para Li *et al.* (2005), um sistema pode ser considerado multiagente quando possui determinadas características, a saber: (1) ambiente; (2) conjunto de agentes; (3) conjunto de objetos; (4) relações que podem existir entre esses agentes e os objetos ou entre os próprios agentes; (5) conjunto de operações que podem ser realizadas.

Jennings e Wooldrige (2001) propuseram uma estrutura típica de sistemas multiagentes que pode ser visualizada na Figura 3.1. Esse sistema possui uma quantidade de agentes que interagem entre eles. Além disso, esses agentes estão imersos em um ambiente, no qual também podem interagir. Cada um deles possui

esferas de influência diferentes, no qual eles podem controlar (ou influenciar) diferentes partes do ambiente. Essas esferas podem coincidir e aumentar as relações de dependência entre os agentes.

Para Lima (2010), a estrutura da Figura 3.1 mostra que adotar uma estratégia orientada a agentes significa decompor o problema em componentes múltiplos e autônomos, os quais podem interagir de forma flexível para alcançar seus objetivos.

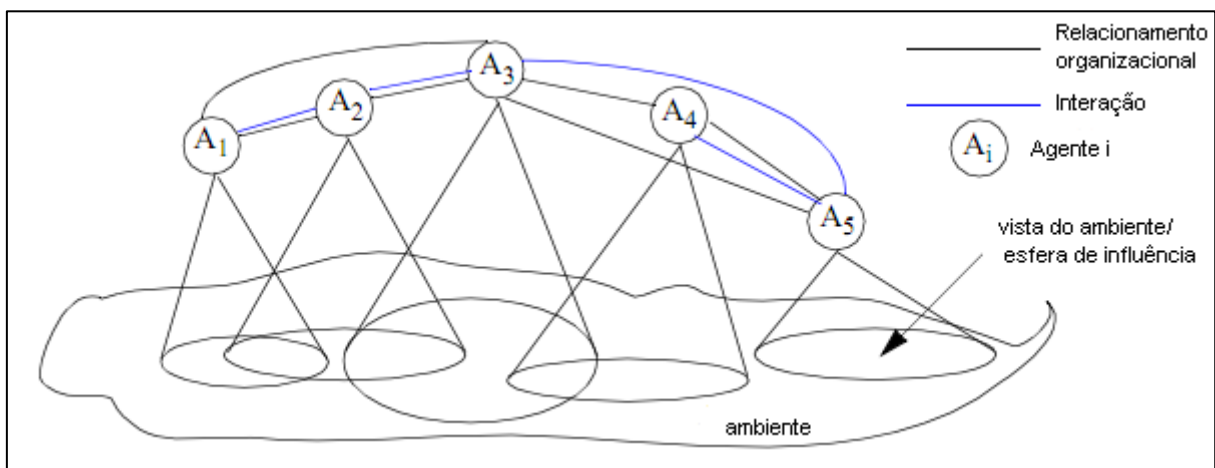


Figura 3.1 – Estrutura típica de sistemas multiagentes
Fonte: Jennings e Wooldrige (2001).

Em relação à classificação desse tipo de sistema, eles podem ser classificados em dois tipos: reativos e cognitivos, embora possam ser híbridos (possuir os dois tipos de agentes) (SILVA, REIS, *et al.*, 1999) (ABREU, 2006).

Os sistemas multiagentes do tipo reativos geralmente são baseados em modelos de organização biológica ou etológica, como formigas, cupins, abelhas (ALVES, 2009). Esta abordagem defende que, em um sistema multiagente, não é necessário que cada agente individual seja inteligente para alcançar um comportamento inteligente global, pois a inteligência está na coletividade. Seu modelo de funcionamento é formado pelo par estímulo-resposta, isto é, ação e reação (LABIDI e LEJOUAD, 1993). Nesses sistemas, não existe uma representação explícita do conhecimento, do ambiente e da memória (ações) dos agentes (ALVES, 2009).

Os sistemas multiagentes cognitivos são baseados em modelos de organização social de sociedades humanas. Os agentes desse tipo podem ter capacidade de raciocinar sobre ações tomadas no passado e planejar futuras ações. Além disso, podem interagir com os demais agentes através de protocolos de

comunicação complexos, tendo capacidade de negociação (ABREU, 2006). Em geral, esse tipo de sistema possui poucos agentes na sociedade (ALVES, 2009). Na Tabela 3.1 é apresentado um breve comparativo de sistemas multiagentes do tipo reativos e cognitivos.

Sistemas de Agentes Reativos	Sistemas de Agentes Cognitivos
Representação implícita	Representação explícita do ambiente
Sem memória (histórico)	Podem ter memória
Operação estímulo/resposta	Agentes complexos
Grande quantidade de agentes	Pequeno número de agentes

Tabela 3.1 – Comparativo entre agentes reativos e cognitivos.
Fonte: Labidi e Lejoaud (1993).

3.3 Arquitetura de Sistemas Multiagentes

A arquitetura de um agente inteligente pode ser reativa, deliberativa ou híbrida (CELAYA, DESROCHERS e GRAVES, 2007) (WU, CAO, *et al.*, 2009).

Segundo Osório *et al.* (2004), uma arquitetura é denominada reativa ou não-deliberativa quando a escolha da ação a ser executada está relacionada de forma direta com a ocorrência de eventos no ambiente. Nessa arquitetura, o controle das ações do agente é realizado a partir de um comportamento do tipo situação – ação (ou estímulo – resposta). O agente age em um espaço de tempo, com base em uma pequena quantidade de informação, no instante em que recebe ou percebe algum sinal ou estímulo do ambiente.

Os agentes modelados a partir de uma arquitetura reativa, denominados reativos ou não-deliberativos, não possuem capacidade de raciocínio e planejamento, e por isso, são consideradas entidades mais simples do que os agentes cognitivos. São agentes baseados em comportamento, definido a partir da situação atual do ambiente e do conhecimento (entrada sensorial) que possuem. Além disso, as ações destes agentes são realizadas em resposta a estímulos oriundos do ambiente (OSÓRIO, MUSSE, *et al.*, 2004).

Já a arquitetura deliberativa ocorre quando a escolha da ação a ser executada pelo agente é realizada a partir de um modelo simbólico do ambiente e de um plano de ações. Essa arquitetura está fundamentada na produção de uma sequência de ações (planos) para alcançar um determinado objetivo. Tais ações

estão baseadas nas hipóteses de que o agente possui um conhecimento do ambiente e de outros agentes. Para isso, é mantida uma representação explícita do conhecimento sobre o ambiente, bem como um histórico das ações passadas (OSÓRIO, MUSSE, *et al.*, 2004).

A arquitetura BDI é um tipo de arquitetura deliberativa e foi concebida por Bratman como uma teoria de raciocínio prático humano (BRATMAN, 1987). Essa teoria é dividida em dois processos: deliberação e raciocínio meio-fim. O processo de deliberação visa deliberar e decidir quais os objetivos um agente quer atingir. Por outro lado, o processo de raciocínio meio-fim destina-se a decidir como os objetivos devem ser alcançados (LOPEZ e MARQUEZ, 2004).

O sucesso do modelo BDI é baseado na sua simplicidade, pois visa simplificar o complexo comportamento humano em uma orientação motivacional (DENNETT, 1987). Isso significa que as causas para agir estão sempre relacionadas com os desejos humanos ignorando outros elementos, como emoções (BRAUBACH, POKAHR e LAMERSDOF, 2005).

Nesse modelo o agente é orientado por suas crenças, desejos e intenções. As crenças representam a informação que um agente possui sobre o ambiente no qual está inserido, além das informações sobre seu próprio estado interno. Isso introduz uma visão pessoal dentro dos agentes, isto é, como eles percebem e refletem sobre o mundo em que fazem parte (BRAUBACH, POKAHR e LAMERSDOF, 2005).

Já os desejos representam as atitudes de motivação dos agentes, que conduzem o curso de suas ações (BRAUBACH, POKAHR e LAMERSDOF, 2005). Os desejos funcionam como metas, isto é, estados que o agente busca obter (LOPEZ e MARQUEZ, 2004).

As intenções são os meios que permitem aos agentes alcançar seus objetivos e reagir a eventos ocorridos. São seus planos operacionais, que não são formados apenas por uma sequência de ações básicas, mas também podem incluir elementos mais abstratos, tais como submetas (BRAUBACH, POKAHR e LAMERSDOF, 2005).

A Figura 3.2 apresenta uma arquitetura abstrata de agentes BDI, onde as elipses representam as principais atitudes mentais dos agentes (crenças, desejos e intenções), enquanto as caixas representam os processos de tomada de decisão da teoria do raciocínio prático humano (BRATMAN, 1987). As linhas tracejadas

representam o fluxo de dados enquanto linhas contínuas representam o fluxo de controle (LOPEZ e MARQUEZ, 2004).

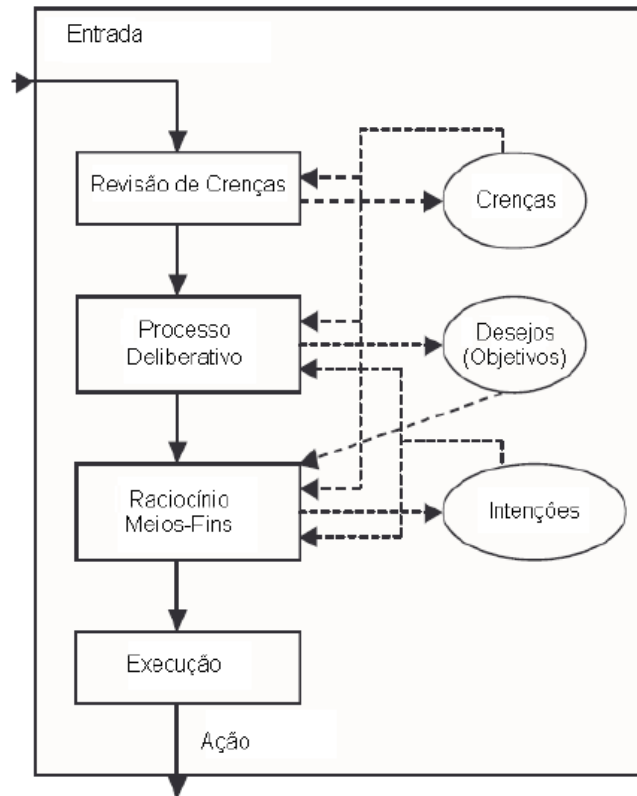


Figura 3.2 – Modelo de Agentes BDI.
Fonte: Lopez e Marquez (2004).

Por fim, arquiteturas híbridas são aquelas cuja a escolha da ação é realizada usando uma combinação entre as técnicas utilizadas em arquiteturas cognitiva e reativa (HEINEN, 2002). Essa arquitetura foi proposta como alternativa para solucionar as deficiências principais das duas arquiteturas anteriores. A arquitetura cognitiva é tipicamente incapaz de reagir rápida e adequadamente perante situações não previstas. Na arquitetura reativa, o agente é incapaz de descobrir alternativas para o seu comportamento quando a situação do mundo diverge bastante de seus objetivos iniciais. Além disso, o agente não possui capacidade de raciocínio e planejamento (OSÓRIO, MUSSE, *et al.*, 2004).

O objetivo é construir um agente atuante em dois subsistemas: o sistema cognitivo, que contém um modelo simbólico do mundo, utilizando planejamento e tomada de decisões, e o sistema reativo, capaz de reagir a eventos que ocorrem no ambiente. Os agentes híbridos são normalmente projetados através de uma

arquitetura hierárquica. Os níveis mais baixos representam o sistema reativo e são usados para a aquisição de informações do ambiente, de outros agentes ou de outras fontes. Os componentes cognitivos, responsáveis pelo planejamento e determinação de objetivos, são usados nos níveis mais altos (BRENNER, ZARNEKOW e WITTING, 1998).

3.4 Aplicações de Sistemas Multiagentes

A área de sistemas multiagentes está em rápida expansão de pesquisa e desenvolvimento (CZARNOWSKI e JEDRZEJOWICZ, 2010) em uma variedade de aplicações, desde sistemas de assistência pessoal até sistemas abertos, complexos, críticos para aplicações industriais (JENNINGS e WOOLDRIDGE, 1998).

Durante a última década, uma série de avanços significativos têm sido realizados, tanto na concepção como na implementação de sistemas multiagentes. Um número de aplicações dessa tecnologia cresce sistematicamente, sendo utilizadas em sistemas reais, apoiando com sucesso áreas industriais e comerciais. Além disso, um grande número de abordagens baseadas em agentes têm sido propostas para solucionar diferentes problemas de otimização (CZARNOWSKI e JEDRZEJOWICZ, 2010).

Segundo Macal e North (2005), o uso de sistemas multiagentes é aplicado em várias áreas, tais como, operações de indústria, mercados de consumo, mercados financeiros artificiais, redes de negócio, mercados de energia elétrica, transporte, usos militares, entre outros. Além disso, os autores citam alguns problemas que se poderiam usar sistemas multiagentes em sua resolução:

- Problemas no qual a complexidade é proveniente da interdependência de diversos fatores;
- Problemas no qual a modelagem é complexa;
- Problemas que envolvam grandes quantidades de dados e que exijam grande poder computacional.

Ainda, de acordo com Tweedale *et al.* (2007), agentes também são utilizados na percepção de situação de ambiente hostis, utilizando a arquitetura BDI. Além disso, para os autores, agentes são adequados para implementar sistemas em diversas áreas, como, sistemas inteligentes de suporte a decisões na área de saúde, militar, comércio eletrônico, gestão de conhecimento, entre outros.

3.5 Considerações Finais

Segundo Jennings e Bussmann (2003), a abordagem orientada a agentes pode melhorar significativamente o processo de desenvolvimento de software para certos tipos de aplicação (médicas, entretenimento, comerciais, industriais, etc). Os agentes permitem uma metáfora natural, isto é, muitos ambientes (incluindo a maioria das organizações) são naturalmente modeladas como sociedades de agentes, seja cooperando para solucionar problemas ou competindo entre si (JENNINGS e WOOLDRIDGE, 1998).

Neste capítulo foi apresentada uma breve revisão bibliografia sobre agentes de software. Além disso, foi mencionado suas características, classificações e categorias, bem como, seus tipos de arquiteturas. Por fim, foram expostas as principais aplicações deste tipo de estrutura de software.

4 MÉTODO DE PESQUISA

Gil (2007, p. 42), define pesquisa como “um processo formal e sistemático de desenvolvimento do método científico”. Ainda, segundo o autor, o objetivo principal da pesquisa é “descobrir respostas para problemas, mediante o emprego de procedimentos científicos”.

Para definir o método de pesquisa, Gil (2007) apresenta algumas classificações existentes de pesquisas, tais como: objetivo da pesquisa, abordagem do problema, procedimentos técnicos, entre outros.

Quanto ao objetivo da pesquisa, esse trabalho é de natureza exploratória. Uma pesquisa exploratória visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito (GIL, 1991). Ainda, este tipo de pesquisa tem como objetivo o aprimoramento de ideias ou a descoberta de intuições e, através de um delineamento bibliográfico, efetuar o levantamento de informações pertinentes ao problema para averiguação do que já foi analisado em relação ao tema (SELLTIZ, JAHODA, *et al.*, 1967).

Quanto à abordagem do problema, uma pesquisa pode ser quantitativa, qualitativa ou multi-método. A distinção das abordagens é feita da seguinte forma:

Uma **abordagem quantitativa** é a que o investigador primeiramente utiliza os pressupostos pós-positivistas para o desenvolvimento do conhecimento (...), emprega estratégias como experimentos e levantamentos e coleta dados por instrumentos pré-determinados que resultem em dados estatísticos. Alternativamente, a **abordagem qualitativa** (...) baseia-se em perspectivas construtivistas ou participativas. Utiliza estratégias de pesquisa como narrativas, fenomenologias, etnografias, estudos de *grounded theory* ou estudos de caso. O pesquisador coleta dados não estruturados e emergentes (...). Finalmente, a **abordagem de multi-método** é aquela em que o pesquisador tende a basear seus pressupostos em campos pragmáticos. Emprega estratégias que envolvem a coleta de dados tanto simultaneamente ou sequencialmente para melhor entender os problemas de pesquisa. A coleta de dados envolve tanto informações numéricas quanto informações textuais (CRESWELL, 2003, p. 19-20).

Este trabalho utiliza a abordagem multi-método, pois combina métodos qualitativos (no levantamento de informações para a modelagem) e quantitativos (para a construção da simulação computacional (ALVES, 2009)).

Quanto aos procedimentos técnicos, na primeira etapa deste trabalho utilizou-se a pesquisa bibliográfica, no qual busca-se informações a partir de material já publicado, constituído principalmente de livros, artigos de periódicos e, atualmente, com material disponibilizado na internet (GIL, 2007). Na segunda etapa deste trabalho utilizou-se pesquisa experimental, que baseia-se na determinação de um objeto de estudo, seleção das variáveis que seriam capazes de influenciá-lo, definição das formas de controle e observação dos efeitos que a variável produz no objeto (GIL, 2007).

A pesquisa experimental foi realizada empregando a técnica de simulação como ferramenta para a realização de experimentos. A simulação permite a geração de cenários, a partir dos quais pode-se orientar o processo de tomada de decisão, proceder análises e avaliações de sistemas e propor soluções para a melhoria de desempenho (SILVA, 2005). Através da simulação é possível testar alternativas de mudanças, sem, contudo, alterar o sistema real. É uma experimentação simples, que pode tomar a forma de questões do tipo “E se fizéssemos esta mudança, qual seria o impacto no sistema?” (OLIVEIRA, 2008).

Desse modo, o presente trabalho busca a melhor compreensão de um problema através da modelagem de um sistema real. Um resumo do método utilizado neste trabalho pode ser visualizado na Tabela 4.1.

Quanto aos objetivos	Pesquisa exploratória
Quanto a abordagem do problema	Abordagem multi-método
Quanto aos procedimentos técnicos	Pesquisa bibliográfica Pesquisa experimental

Tabela 4.1 – Método da Pesquisa

4.1 Etapas do Desenvolvimento

Este trabalho será desenvolvido de acordo com os seguintes passos:

- Caracterização do modelo a ser simulado;
- Escolha do ambiente de simulação;
- Construção da ferramenta de simulação multiagente;
- Elaboração dos experimentos;

- Validação do modelo gerado, através da aplicação de um conjunto de experimentos de simulação sobre a ferramenta desenvolvida.

Os passos de desenvolvimento do trabalho serão descritos detalhadamente nos próximos capítulos.

5 CONSTRUÇÃO DO MODELO DE SIMULAÇÃO

Este capítulo apresenta a construção do modelo de simulação multiagente de alocação de tarefas a pessoas em projetos de software. Primeiramente é apresentado o processo de modelagem utilizado para a construção da simulação e do modelo em si. Logo após, apresenta-se as premissas e os atributos do modelo, bem como as equações que o regem. Além disso, são expostos os comportamentos dos agentes, assim como a comunicação entre eles. Por fim, é apresentado o framework utilizado para desenvolver a simulação.

5.1 Processo de Modelagem

Para Streit (2006, p. 86), “a modelagem consiste no processo de representação de um sistema sob análise no mundo real para um modelo computacional, capaz de assumir um comportamento semelhante ao sistema”. Lin (2007, p. 27) descreve um modelo conceitual como “uma forma de abstração que permite modelar estruturas de dados naturais, independente de sua organização física. Este modelo define regras de acordo com os dados que necessitam ser estruturados”. Ainda, o autor afirma que uma modelagem orientada a agentes é uma forma de modelo conceitual, onde a maioria das entidades é definida como agentes, representando papéis e responsabilidades.

Para modelar um sistema multiagente devem-se considerar vários aspectos, tais como comunicação, interação, arquitetura (CELAYA, DESROCHERS e GRAVES, 2007). Macal e North (2005) propõem um processo de modelagem de simulação de sistemas multiagentes composto pelos seguintes passos:

- a) Identificar o propósito do modelo, as questões que o modelo pretende responder e os usuários potenciais;
- b) Identificar os agentes do modelo e definir sua teoria comportamental;
- c) Identificar o relacionamento dos agentes e definir sua teoria de interação;
- d) Definir a plataforma de desenvolvimento de sistemas multiagentes e uma estratégia de modelagem;
- e) Obter os dados necessários relacionados aos agentes;
- f) Validar o modelo de comportamento dos agentes, bem como o modelo como um todo;

g) Executar o modelo e analisar os resultados.

Para realizar a modelagem do sistema, existem várias técnicas propostas na literatura, tais como, AUML (ODELL, PARUNAK e BAUER, 2000), Prometheus (PADGHAM e WINIKOFF, 2002), MaSE (WOOD e DELOACH, 2000), Tropos (BRESCIANI, PERINI, *et al.*, 2002), entre outras. Para modelar os agentes do sistema, foi utilizado o *framework* proposto por Jo *et al* (2004) com foco na arquitetura BDI.

5.2 Propósito do Modelo

O modelo em questão tem como propósito auxiliar gestores no processo de tomada de decisão em relação à alocação de pessoas em atividades de projetos de software. As questões na qual o modelo pretende responder podem ser visualizadas em detalhes no capítulo 1 desse documento.

5.3 Identificação dos Agentes

Para identificar os agentes participantes do modelo, é necessário escolher uma teoria comportamental (MACAL e NORTH, 2005). Segundo Lopez e Marquez (2004) e Streit (2006), um dos mais conhecidos modelos de arquitetura deliberativa é a BDI¹. Esta arquitetura foi apresentada na seção 3.3 com maiores detalhes.

Na próxima seção será apresentado um método para modelagem de agentes BDI proposto por Jo *et al.* (2004), utilizado para modelar os agentes desse trabalho.

5.4 Modelagem BDI

O processo usado nessa abordagem para descobrir as crenças, desejos e intenções pode ser visualizada na Figura 5.1. Os passos do processo serão apresentados nas próximas seções.

¹ BDI: Significa Beliefs (crenças), Desires (desejos) e Intentions (intenções).

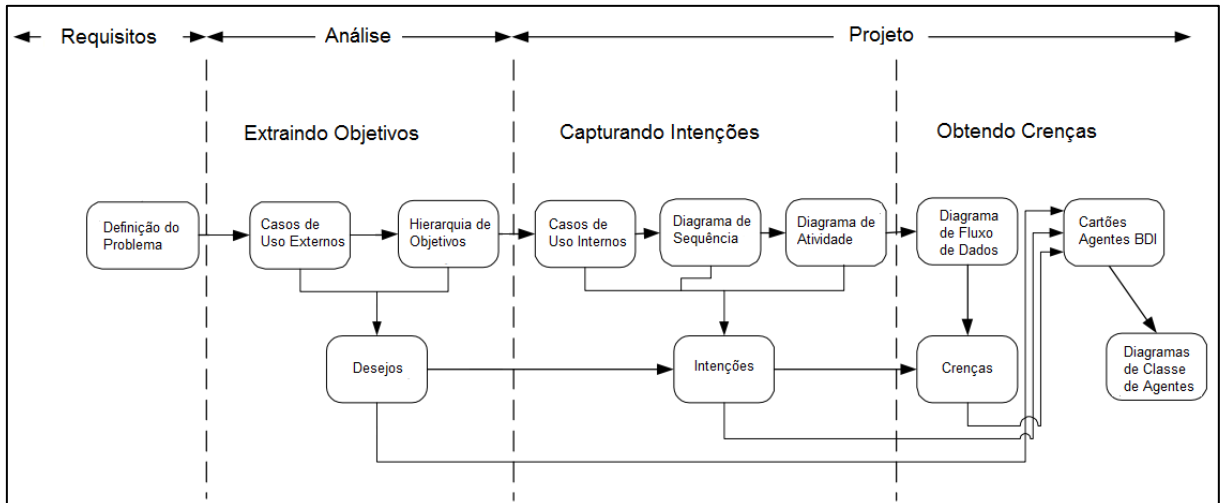


Figura 5.1 – Processo de modelagem do Framework BDI
 Fonte: Jo et al (2004, p. 290).

5.4.1 Casos de Uso Externos

Os casos de uso externos visam mostrar como entidades externas interagem e usam o sistema em questão. Os objetivos podem ser extraídos do que o sistema está tentando atingir e geralmente permanecem constantes durante todo o processo de análise e projeto (JO, CHEN e CHOI, 2004).

A primeira tarefa dessa fase é definir o objetivo principal do sistema. Além de capturar seus objetivos, também é possível descrever suas interações com o ambiente, descobrindo atores externos envolvidos (JO, CHEN e CHOI, 2004).

Durante essa fase é possível identificar os serviços que o sistema oferece, do ponto de vista externo; não são descritos processos internos, componentes, ou projeto do sistema. Através dessa atividade é possível definir quem (ator) realiza algo (interação) com o sistema, com o propósito (objetivo), sem lidar com aspectos internos do sistema (JO, CHEN e CHOI, 2004).

O objetivo global desse sistema é gerar o cronograma do projeto, através do controle da atribuição das atividades aos membros participantes. Essa atribuição pode ser realizada de acordo com 3 estratégias: preferência por prazo, custo ou desenvolvimento da equipe do projeto.

Na preferência por prazo os membros participantes do projetos buscam realizar atividades com o intuito de alcançar o melhor prazo possível de entrega do projeto. Já a preferência por custo trata-se da busca pelo menor custo possível de execução do projeto. O custo de execução do projeto nesse caso refere-se exclusivamente ao custo da mão-de-obra envolvida na realização das tarefas. Por

fim, a terceira estratégia busca considerar os desejos dos membros participantes em desenvolver habilidades profissionais através da execução de atividades.

Na Figura 5.2 é possível visualizar o caso de uso externo do sistema, onde o principal objetivo é gerar o cronograma do projeto. Existem três funcionalidades ou serviços que permitem realizar este objetivo. Tais funcionalidades são detalhadas na próxima seção.

Caso de Uso Externo
Gerar cronograma do projeto.
<ol style="list-style-type: none"> 1. Controle de alocação de atividades. 2. Cálculo de duração do projeto. 3. Cálculo de custo do projeto.

Figura 5.2 – Caso de uso externo
Fonte: elaborada pela autora.

5.4.2 Hierarquia de Objetivos

Neste passo se busca mapear os objetivos derivados dos casos de uso externos em uma hierarquia de objetivos. Para isso, é necessário reconhecer o objetivo principal do sistema e dividi-lo em objetivos menores. A partir dessa hierarquia é possível definir melhor os objetivos do sistema (desejos) (JO, CHEN e CHOI, 2004). Na Figura 5.3 pode-se visualizar o objetivo principal “Gerar cronograma” decomposto em três sub-objetivos. O primeiro sub-objetivo refere-se a controlar o processo de alocação da atividades. O gestor é responsável por informar aos membros participantes do projeto as atividades disponíveis e os responsáveis por executá-las. O gestor é responsável por todas as atualizações de estado das atividades, bem como, tem conhecimento de detalhes de sua execução, tais como os responsáveis pela tarefa e o prazo de execução, entre outros. O segundo e o terceiro sub-objetivos referem-se aos cálculos necessários para a geração do cronograma do projeto, tais como, o prazo e o custo.

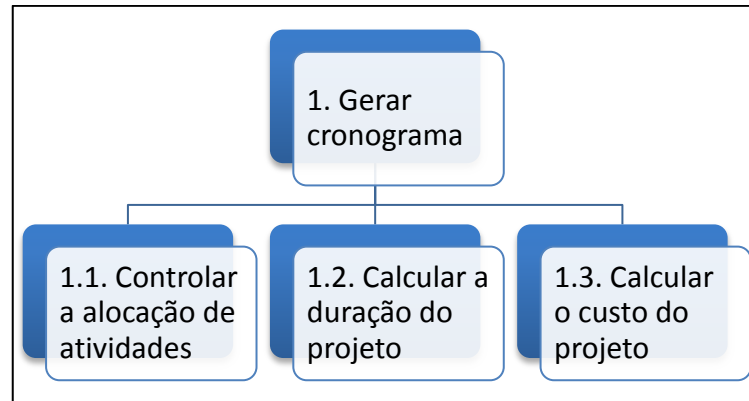


Figura 5.3 – Hierarquia de objetivos - Gestor
Fonte: elaborada pela autora.

Na Figura 5.3 são apresentados os objetivos do sistema sob o ponto de vista do gestor. Os objetivos sob a perspectiva dos membros participantes do projeto podem ser visualizados na Figura 5.4. Tais objetivos possuem total relação com os objetivos previamente mostrados e são separados para facilitar a construção do sistema e visualizar melhor as comunicações entre os agentes.

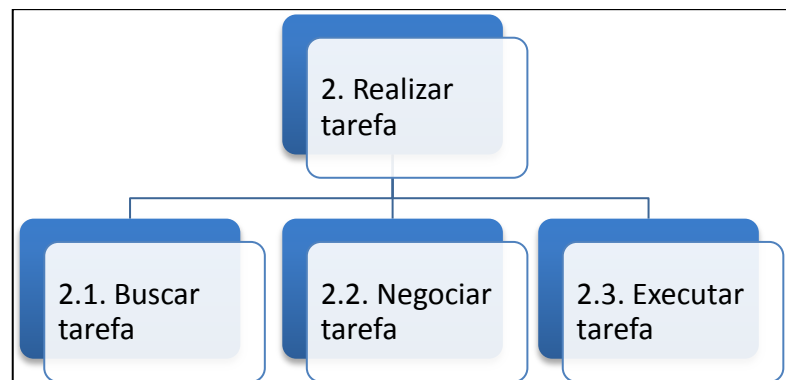


Figura 5.4 – Hierarquia de objetivos – Membro
Fonte: elaborada pela autora.

Na Figura 5.4, o objetivo “Realizar tarefa” foi subdividido em três sub-objetivos. O primeiro sub-objetivo refere-se ao processo de buscar tarefa realizado pelo membro, no qual o mesmo solicita ao gestor informação sobre as tarefas disponíveis existentes e seleciona a que possuir maior compatibilidade de acordo com as suas características e a estratégia de configuração do sistema. O segundo sub-objetivo refere-se ao processo de negociação de tarefa com outros membros do projeto, caso a atividade desejada esteja em posse de outra pessoa. O terceiro sub-

objetivo trata da execução da atividade em si, permitindo ao indivíduo ganho de experiência (aperfeiçoamento de suas habilidades).

5.4.3 Casos de Uso Internos

Os casos de uso internos estão preocupados com a interação entre os elementos internos do sistema. O propósito dessa fase é identificar as intenções do sistema. Esse processo é feito através da decomposição dos casos de uso externos em cenários (JO, CHEN e CHOI, 2004).

Os casos de uso internos relativos ao gestor são apresentados na Tabela 5.1, Tabela 5.2 e Tabela 5.3, onde é possível visualizar de forma detalhada o cenário principal dos processos descritos na Figura 5.3.

Caso de Uso	1.1. Controlar a alocação de atividades
Objetivo	Gerenciar o processo de alocação de atividades
Ator Primário	Gestor
Stakeholders	Membro do projeto
Pré-condições	<ul style="list-style-type: none"> • Projeto com atividades • Membros com capacidade de executar as atividades disponíveis do projeto
Pós-condições	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O membro solicita lista de atividades ao gestor. 2. O gestor envia a lista de atividades do projeto. 3. O membro seleciona a atividade desejada e notifica ao gestor o interesse em realizar essa tarefa. 4. O gestor notifica disponibilidade da tarefa ao membro. 5. O membro executa a atividade e avisa ao gestor o término da execução.

	6. O gestor armazena histórico da execução.
--	---

Tabela 5.1 – Caso de Uso Interno “Controlar a alocação da Atividade”
Fonte: elaborada pela autora.

Caso de Uso	1.2. Calcular a duração do projeto
Objetivo	Realizar o cálculo de duração do projeto com o intuito de gerar o cronograma.
Ator Primário	Gestor
Stakeholders	Nenhum
Pré-condições	<ul style="list-style-type: none"> • Término da execução de todas as atividades do projeto • Histórico da execução das atividades
Pós-condições	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O gestor organiza as atividades de acordo com sua ordem de execução. 2. O gestor calcula a duração total do projeto.

Tabela 5.2 – Caso de Uso Interno “Calcular a duração do projeto”
Fonte: elaborada pela autora.

Caso de Uso	1.3. Calcular o custo do projeto
Objetivo	Realizar o cálculo do custo do projeto.
Ator Primário	Gestor
Stakeholders	Nenhum
Pré-condições	<ul style="list-style-type: none"> • Término da execução de todas as atividades do projeto • Histórico da execução das atividades
Pós-condições	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O gestor consulta os custos de mão-de-obra por hora de execução de atividades. 2. O gestor calcula o custo total do

	projeto.
--	----------

Tabela 5.3 – Caso de Uso Interno “Calcular o custo do projeto”
Fonte: elaborada pela autora.

Os casos de uso internos relativos ao membro são apresentados na Tabela 5.4, Tabela 5.5 e Tabela 5.6, onde é possível visualizar de forma detalhada o cenário principal dos processos descritos na Figura 5.4.

Caso de Uso	2.1. Buscar tarefa
Objetivo	Buscar atividade compatível para execução.
Ator Primário	Membro
Stakeholders	Gestor
Pré-condições	<ul style="list-style-type: none"> • Projeto com atividades disponíveis • Capacidade de executar atividades disponíveis do projeto
Pós-condições	Selecionar a atividade mais compatível.
Cenário Principal	<ol style="list-style-type: none"> 1. Membro solicita ao gestor a lista de atividades disponível. 2. Membro escolhe a atividade com maior compatibilidade, de acordo com a configuração do sistema. 3. Membro avisa ao gestor o interesse em realizar dada atividade. 4. O gestor reserva a realização da atividade a este membro.
Cenário Alternativo	<i>Atividade desejada não disponível</i> 3.1. Executar processo “2.2. Negociar Tarefa”.

Tabela 5.4 – Caso de Uso Interno “Buscar tarefa”
Fonte: elaborada pela autora.

Caso de Uso	2.2. Negociar tarefa
Objetivo	Negociar uma tarefa desejada com outro membro, definindo quem vai executá-la.

Ator Primário	Membro
Stakeholders	Outros membros, Gestor
Pré-condições	<ul style="list-style-type: none"> • Tarefa desejada estar pré-allocada a outro membro. • Capacidade de executar a atividade em questão.
Pós-condições	Nenhuma.
Cenário Principal	<ol style="list-style-type: none"> 1. Gestor informa ao membro quem é o responsável por realizar a tarefa desejada. 2. Membro solicita a execução da tarefa a outro membro. 3. Membro recebe a execução da tarefa e avisa o gestor sobre a troca de responsável.
Cenário Alternativo	<p><i>Membro não possui maior compatibilidade</i></p> <ol style="list-style-type: none"> 3.1. Membro não recebe a execução da tarefa, pois o outro membro possui maior compatibilidade. 3.2. Membro marca tarefa como inativa. 3.3. Membro volta ao processo “2.1. Buscar Tarefa”.

Tabela 5.5 – Caso de Uso Interno “Negociar tarefa”
 Fonte: elaborada pela autora.

Caso de Uso	2.3. Executar tarefa
Objetivo	Executar tarefa alocada.
Ator Primário	Membro
Stakeholders	Gestor
Pré-condições	<ul style="list-style-type: none"> • Tarefa desejada estar alocada ao membro em questão. • Período de negociação terminado.
Pós-condições	Ganho de experiência.
Cenário	1. Membro avisa ao gestor início de

Principal	<p>execução da atividade.</p> <ol style="list-style-type: none"> 2. Gestor muda estado da atividade para “alocada”. 3. Membro executa a atividade. 4. Membro avisa ao gestor o término da atividade. 5. Gestor muda estado da atividade para “concluída”. 6. Membro atualiza sua experiência com o novo ganho. 7. Membro busca novas tarefas para realizar através do processo “2.1. Buscar Tarefa”.
------------------	--

Tabela 5.6 – Caso de Uso Interno “Executar tarefa”
 Fonte: elaborada pela autora.

5.4.4 Diagramas de Sequência

Os diagramas de sequência têm por objetivo ilustrar a sequência de eventos e o relacionamento entre papéis (um papel inclui um objetivo particular ou um conjunto de objetivos e um conjunto de intenções). Esses diagramas pretendem mostrar como os papéis se comunicam um com o outro durante a execução do processo. Esses papéis são candidatos a agentes (JO, CHEN e CHOI, 2004). Na Figura 5.5, Figura 5.6 e Figura 5.7 são apresentados os diagramas de sequência correspondentes aos casos de uso relacionados ao membros participantes de um projeto.

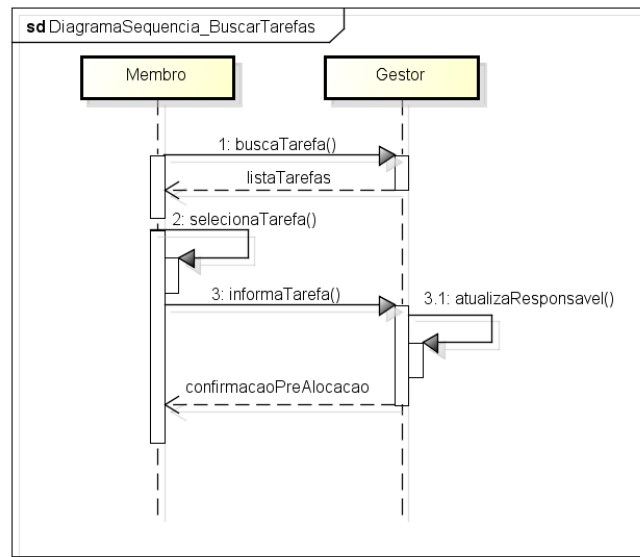


Figura 5.5 – Diagrama de Sequência Buscar Tarefas
Fonte: elaborada pela autora.

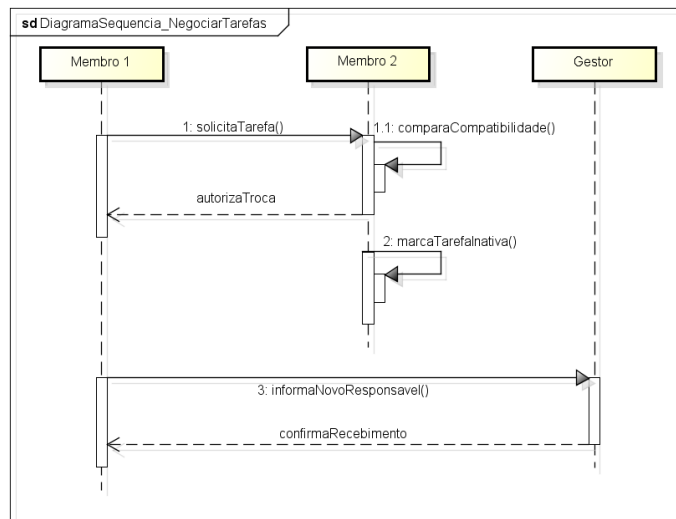


Figura 5.6 – Diagrama de Sequência Negociar Tarefas
Fonte: elaborada pela autora.

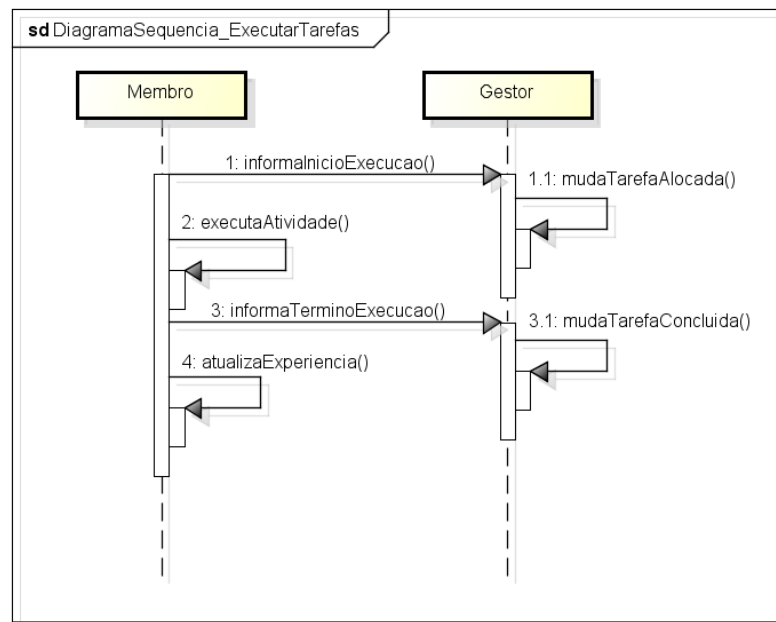


Figura 5.7 – Diagrama de Sequência Executar Tarefas
Fonte: elaborada pela autora.

5.4.5 Diagramas de Atividades de Agente

O diagrama de atividades de agente apresenta as operações e os eventos gerados por agentes. Semelhante ao diagrama de atividades UML, esse diagrama descreve todos os eventos disparado pelos agentes potenciais (JO, CHEN e CHOI, 2004). Na Figura 5.8 pode ser visualizado um diagrama de atividades geral que engloba os casos de uso e diagramas de sequência supracitados.

5.4.6 Lista de Crenças do Agente

As crenças dos agentes são um conjunto de dados que descrevem o estado do ambiente que os cerca. Elas são o conhecimento que as intenções utilizam para cumprir seus objetivos (desejos). A fim de encontrar esses dados, esse método aplica um diagrama de fluxo de dados, o qual mostra o fluxo dos dados de entidades externas ao sistema, e como tais dados se movem de um processo ao outro, bem como seu armazenamento (JO, CHEN e CHOI, 2004).

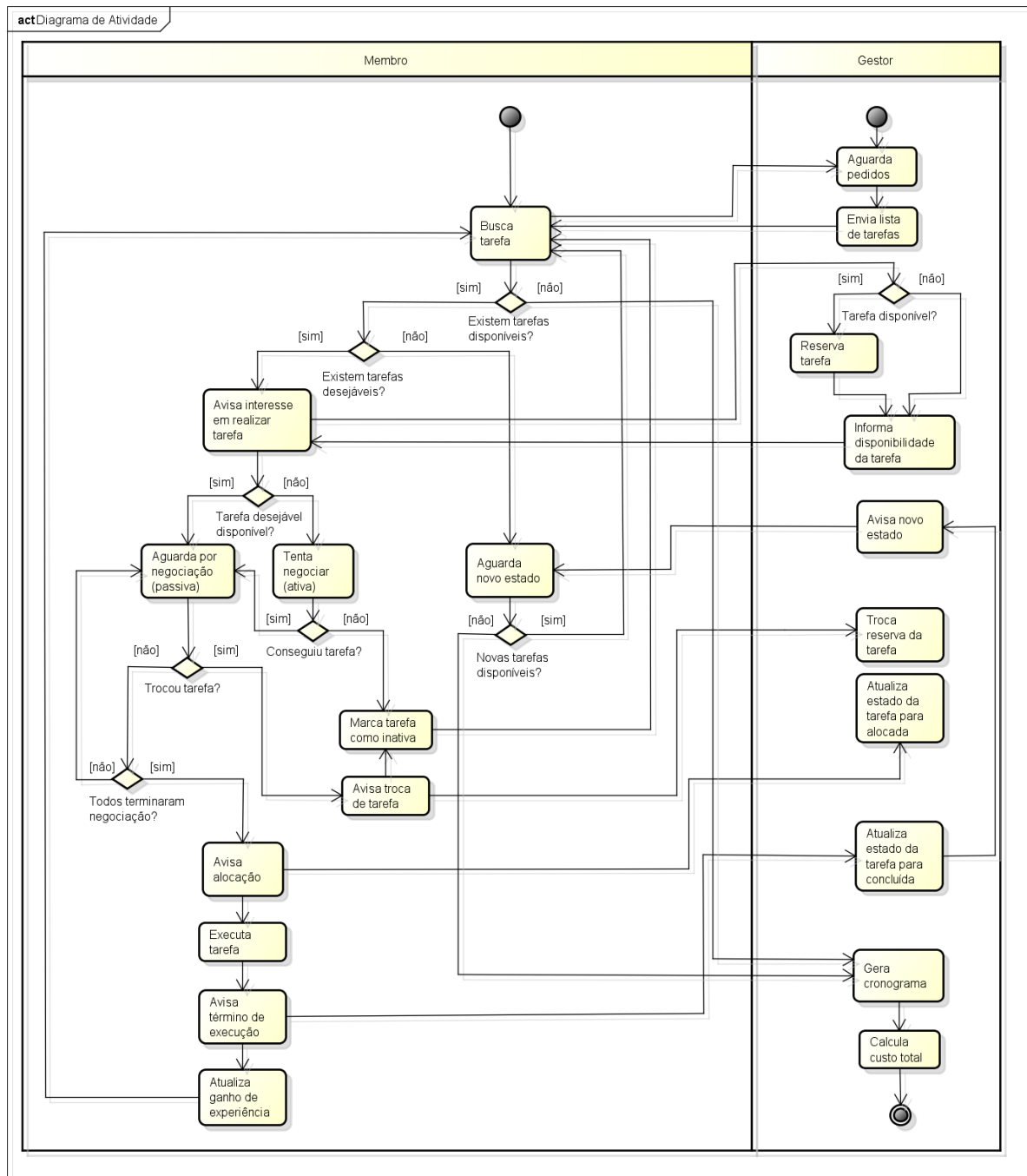


Figura 5.8 – Diagrama de Atividades de Agente
Fonte: elaborada pela autora.

O propósito principal desse diagrama é descobrir os dados envolvidos no processo. Sua notação é composta de círculos, setas, linhas e retângulos. Os círculos representam os processos, que recebem entradas e as transformam em saídas. Já as setas que entram ou saem de um processo se referem aos fluxos, e são utilizadas para mostrar o movimento de informações de um ponto a outro do sistema. Os depósitos são apresentados no modelo como duas linhas paralelas e

representam dados em repouso, como uma base de dados para leitura ou escrita. Por fim, os terminadores apresentados no diagrama como retângulos representam entidades externas com as quais o sistema se comunica.

Na Figura 5.9 pode ser visualizado o diagrama de fluxo de dados com os principais processos do sistema. Os dados acompanham as setas, movendo-se dentro do sistema. Os depósitos Tarefas e Membros são bases de dados que guardam informações das tarefas e dos membros, necessárias para realizar os processos de seleção, negociação, comparação de compatibilidade, execução e cálculo de experiência. O gestor de projetos é uma entidade externa ao sistema que recebe o cronograma do projeto, bem como o cálculo do custo, após a execução da simulação.

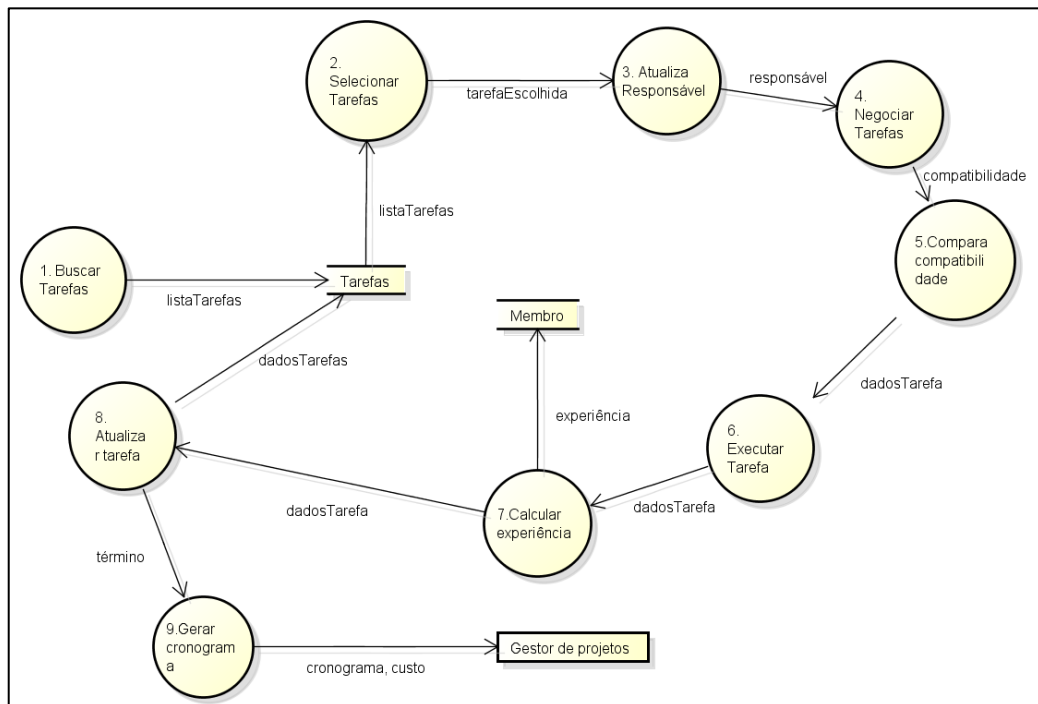


Figura 5.9 – Diagrama de Fluxo de Dados
Fonte: elaborada pela autora.

De acordo com a Figura 5.9, pode-se identificar como crenças do agente membro o conhecimento sobre si próprio, como sua identificação, habilidades, entre outros. Além disso, também possui conhecimento sobre a existência de tarefas disponíveis e outros agentes membros, bem como sobre a compatibilidade de membros e tarefas e a experiência que uma atividade entrega. O agente gestor tem conhecimento sobre os agentes membros, as tarefas e seus responsáveis, sobre

dados do projeto, como cronograma e custo. Na Tabela 5.7 apresenta-se as crenças dos agentes membro e gestor.

Agente Membro	Tarefas disponíveis Universo de agentes Compatibilidade entre tarefas Experiência provida por uma tarefa
Agente Gestor	Tarefas Universo de agentes Responsáveis pelas tarefas Dados do projeto

Tabela 5.7 – Crenças dos agentes

Fonte: elaborada pela autora.

5.4.7 Cartões de Agentes BDI

Os cartões de agentes BDI apresentam todas as entidades de um agente desse tipo: agentes, crenças, desejos e intenções. Nessa seção todos os dados previamente apresentação são organizados em grupos baseados nos objetivos que se deseja alcançar (JO, CHEN e CHOI, 2004). Na Tabela 5.8 é possível visualizar as crenças, desejos e intenções dos agentes membros e gestor.

Agente	Membro	Gestor
Crenças	<ul style="list-style-type: none"> • Conhecimento sobre si próprio • Tarefas disponíveis • Universo de agentes • Compatibilidade entre tarefas • Experiência provida por uma tarefa 	<ul style="list-style-type: none"> • Tarefas • Universo de agentes • Responsáveis pelas tarefas • Dados do projeto
Desejos	Concluir atividades de acordo com a característica do sistema	Coordenar a alocação de tarefas Gerar resultados finais (cronograma, custo)
Intenções	<ul style="list-style-type: none"> • escolherAtividade() • solicitarListaAtividades() • solicitarResponsavelAtividade() • avisarInicioExecucao() • avisarTerminoExecucao() 	<ul style="list-style-type: none"> • enviarListaAtividades() • coordenarAlocaçãoAtividades() • gerarCronograma() • atualizarListaAtividades() • calcularCustoProjeto()

	<ul style="list-style-type: none"> • solicitarDadosAtividade() • calcularExperienciaGanha() • calcularCompatibilidade() • negociacaoAtiva() • negociacaoPassiva() 	
--	--	--

Tabela 5.8 – Cartões de Agentes BDI
Fonte: elaborada pela autora.

O produto final dessa fase é o diagrama de classes de Agente. As crenças geram os atributos e as intenções os métodos. Esse diagrama pode ser visualizado na Figura 5.10. É importante salientar que o diagrama apresentado nessa fase é apenas conceitual que não engloba todas as classes a serem implementadas no sistema. Para a implementação foram criadas outras entidades, como Habilidade e Atividade.

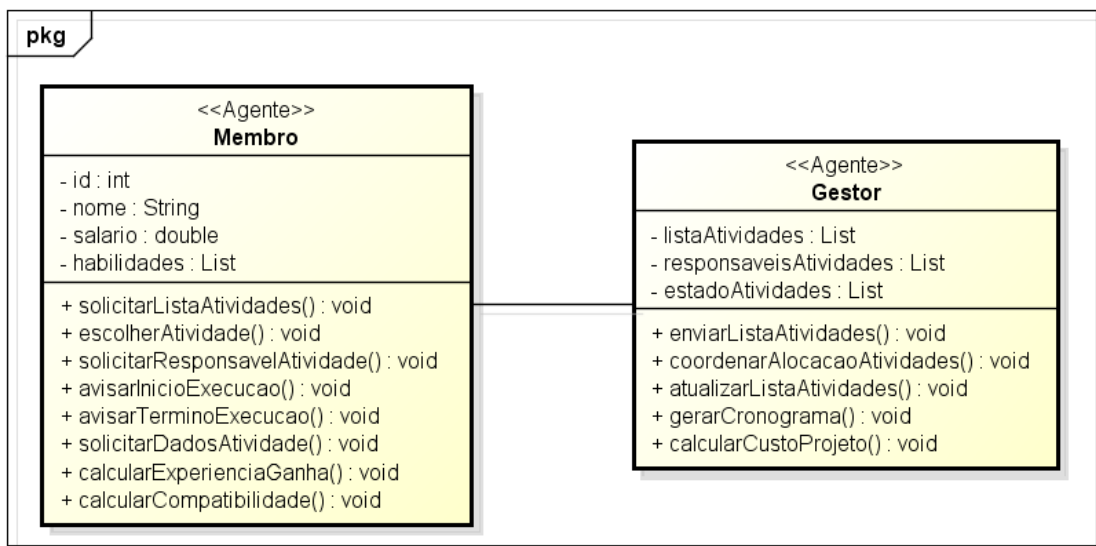


Figura 5.10 – Diagrama de Classes Conceitual
Fonte: elaborada pela autora.

5.5 Premissas do Modelo

As premissas do modelo de simulação consideradas neste trabalho são:

- a) Este modelo apenas considera a alocação de recursos humanos. Não são considerados neste modelo outros tipos de recursos.
- b) Quando o modelo se refere a desenvolvimento de pessoas, utiliza-se o método de aprendizagem prática citado no capítulo 2 deste trabalho.

- c) Não há realização de atividades em conjunto. Uma tarefa é sempre realizada por apenas uma pessoa.
- d) Os elementos que compõem uma atividade foram definidos de acordo com a seção 5.5.1.
- e) Os atributos de uma competência foram definidos de acordo com a seção 5.5.2.
- f) Os atributos de um membro participante de um projeto foram definidos de acordo com a seção 5.5.3.
- g) O nível de experiência de cada pessoa afeta a escolha da tarefa, bem como a velocidade de execução da mesma. O cálculo da experiência é apresentado com maior detalhe na seção 5.5.4.
- h) Os comportamentos apresentados pelos agentes nesse modelo são explicados na seção 5.5.5.
- i) Existem dois tipos de comunicação entre agentes: entre gestor e membro e entre um membro e outro. Essa comunicação é detalhada na seção 5.5.6.
- j) O processo de alocação de recursos humanos consiste em designar às pessoas participantes a atividade “mais adequada” ao seu perfil. Primeiramente busca-se definir atividade. O termo “mais adequada” depende da configuração do sistema, ou seja, dos elementos prazo e aprendizado. Os modelos de configuração do sistema são descritos com maiores detalhes na seção 5.5.7.
- k) O decorrer do tempo na simulação funciona de forma diferente aos modelos tradicionais. O sistema calcula o término de cada tarefa e avança para esse momento para distribuição de novas atividades e cálculo de experiência ganha. O período de tempo destinado à distribuição de tarefas não é considerado como tempo decorrido na simulação.
- l) No modelo de configuração por aprendizado, a escolha de tarefas se dá através da tarefa que fornece mais experiência por dia (curto prazo) na competência que se deseja desenvolver.

5.5.1. Atributos de uma Atividade

Para definir que elementos compõem uma atividade, foram considerados vários trabalhos relacionados. Segundo os trabalhos de [1] (FATIMA e WOOLDRIDGE, 2001), [2] (SCHNAIDER, 2003), [3] (BARRETO, 2005), [4] (COELHO, 2007), [5] (XIAO, WANG, *et al.*, 2008) e [6] (RAMOS, 2009), os elementos de atividade encontrados podem ser visualizados na Tabela 5.9.

Elementos da Atividade	Descrição
ID	Refere-se à identificação da atividade
Nome da atividade	Nome dado à atividade
Tipo	Refere-se ao tipo de atividade, por exemplo, no caso de software, poderia ser “Requisitos”
Atividades Predecessoras	Conjunto de atividades cuja realização deve ser terminada antes de executar a atividade em questão
Tamanho	Medido pelo tamanho de produto de trabalho gerado pela atividade
Tipo da habilidade	Usado para descrever certo aspecto da habilidade, como por exemplo, “linguagem de programação”
Nome da habilidade	Nome que especifica o tipo da habilidade, como “Java”.
Nível requerido	Nível necessário para executar atividade, que pode ser medido de várias formas. Por exemplo, para linguagem de programação Java, 3 anos de experiência
Nível máximo	Valor máximo que uma habilidade pode alcançar
Data inicial	Data que a atividade começa/deve começar a ser executada
Data de entrega	Data em que uma atividade deve estar pronta
Orçamento Requerido	Orçamento que uma atividade dispõe
Prioridade	Ordem de prioridade de execução de uma dada atividade
Duração	Tempo de duração de uma dada atividade (Duração = Data_Entrega – Data_Inicial)
Esforço total previsto	Esforço estimado para uma determinada atividade

Tabela 5.9 – Elementos de uma atividade
Fonte: elaborada pela autora.

Na Tabela 5.10 pode-se visualizar a comparação desses elementos em cada trabalho. As fontes aparecem na ordem apresentada no parágrafo anterior. É importante salientar que, em alguns trabalhos, não estava claro todos os atributos que foram utilizados.

Elementos da Atividade	[1]	[2]	[3]	[4]	[5]	[6]
ID	N	N	N	N	S	N
Nome da atividade	N	S	S	S	N	S
Tipo	N	N	N	N	S	N
Atividades Predecessoras	N	N	S	S	S	N
Tamanho	N	N	N	N	S	N
Tipo da habilidade	S	S	N	N	S	S
Nome da habilidade	N	N	S	N	S	S
Nível requerido	S	S	S	S	S	S
Nível máximo	N	N	S	N	N	N
Data inicial	N	S	S	S	S	S
Data de entrega	S	S	S	S	S	S
Orçamento Requerido	N	N	N	N	S	N
Prioridade	S	N	N	N	N	N
Duração	S	N	N	N	N	S
Esforço total previsto	N	N	N	S	N	N

Tabela 5.10 – Comparação de elementos de atividade
Fonte: elaborada pela autora.

O modelo elaborado considerou os elementos de atividade citados na Tabela 5.10, escolhendo os mais relevantes para o trabalho em questão. Além disso, foram adicionados outros atributos conforme necessidade. Os atributos utilizados nesse trabalho podem ser visualizados na Tabela 5.11. Cada tarefa tem um identificador único que garante a alocação dela a apenas uma pessoa. Além disso, cada atividade possui um nome e uma categoria segundo a sua natureza. Essa categorização serve para identificar que área de conhecimento essa atividade está situada, que permite comparar com a compatibilidade de uma pessoa. Também são definidas as datas de início e entrega estimadas para a atividade que servem para calcular sua duração, bem como as datas de início e término da execução da simulação. Essas datas não são necessariamente iguais, pois dependem de certos fatores, tais como, configuração da simulação, experiência do colaborador e tempo de execução de tarefas predecessoras. A atividade também possui os requisitos de competências necessárias que uma pessoa necessita possuir para ser capaz de completar essa tarefa. Por fim, para controlar o estado da tarefa, foram definidas

quatro condições: disponível, onde a tarefa se encontra disponível para qualquer pessoa capaz de executá-la; alocada, que significa que uma tarefa já está sendo executada por um membro do projeto; bloqueada, onde a tarefa por algum motivo (por exemplo, é uma atividade sucessora) não pode ser atribuída a nenhum participante; e, por fim, concluída, onde a atividade já foi terminada por alguém e não está mais na lista de atividades a serem executadas.

Atributos da Atividade	Descrição
Id	Identificador da atividade.
Nome	Nome dado a atividade.
Tipo	Espécie de categoria da atividade
Atividades Predecessoras	Conjunto de atividades cuja realização deve ser terminada antes de começar a execução da atividade em questão.
Data Inicial	Data programada para iniciar a atividade.
Data de Entrega	Data planejada para entregar a atividade concluída.
Data de Início da Execução	Data em que efetivamente começou a execução da atividade.
Data de Término da Execução	Data em que efetivamente terminou a execução da atividade.
Duração	Tempo estimado para a execução de uma atividade.
Competências	Conjunto de competências que são necessárias para a execução da atividade (refere-se à seção 5.5.2).
Estado	Estado da atividade, que pode ser: (1) disponível; (2) alocada; (3) bloqueada; (4) concluída.

Tabela 5.11 – Elementos de uma Atividade
Fonte: elaborada pela autora.

5.5.2. Atributos de uma Competência

O modelo de competências definido nesse trabalho possui atributos semelhantes ao *framework* de competências para Engenheiros de Software (RIVERA-IBARRA, RODRIGUEZ-JACOBO e SERRANO-VARGAS, 2010). Nesse modelo as competências são classificadas em três categorias e influenciam principalmente o atributo área, apresentado na Tabela 5.12. São elas:

- Técnicas: referem-se às atividades a serem desempenhadas, como por exemplo, conhecimento em análise de requisitos.

- Sociais: referem-se às habilidades dos profissionais em interagir em ambientes de trabalho, tais como, trabalho em equipe.
- Pessoais: descreve características particulares de um profissional, como a capacidade de aprender sozinho.

Na Tabela 5.12 são descritos os atributos de uma competência, necessários para definir os requisitos de uma atividade e as competências que uma pessoa possui. Assim como a atividade, a competência possui um identificador único, um nome e uma área que categoriza esse elemento. Além disso, a competência possui um nível exigido e experiência, apresentado com maiores detalhes na seção 5.5.4.

Com respeito ao nível, segundo Gaylor e Ried (2006) e Melo (2007), a política de definição desses níveis hierárquicos depende da organização. Para os autores, o modelo mais frequentemente praticado nas empresas é o nível por tempo de trabalho e se divide em Júnior, Pleno, Sênior e *Master*. Como tal dado é proveniente das normas da empresa, esse parâmetro pode ser configurável na simulação. Nas simulações desse trabalho foi utilizada a configuração apresentada na Tabela 5.13.

Atributos da Competência	Descrição
Id	Identificador da competência.
Nome	Nome dado à competência.
Área	Área da competência, por exemplo, "Programação".
Nível	Nível da competência, que pode ser: (1) Júnior; (2) Pleno; (3) Sênior; (4) Master.
Experiência	Valor numérico que descreve a quantidade de experiência de uma pessoa.

Tabela 5.12 – Atributos da Competência
Fonte: elaborada pela autora.

Nível	Tempo (anos)
Júnior	0
Pleno	1
Sênior	3
Master	5

Tabela 5.13 – Níveis de experiência
Fonte: elaborada pela autora.

5.5.3. Atributos de uma Pessoa

Os atributos relacionados aos recursos humanos encontrados na literatura, de acordo com (FATIMA e WOOLDRIDGE, 2001), (SCHNAIDER, 2003), (BARRETO, 2005), (COELHO, 2007), (XIAO, WANG, *et al.*, 2008) e (RAMOS, 2009), podem ser visualizados na Tabela 5.14. Os nomes utilizados não são necessariamente os mesmos, mas indicam o mesmo atributo.

Atributos	Descrição
ID	Identificação do recurso humano
Tipos de Atividades	Tipos de atividade que a pessoa é capaz de executar
Tipo de Habilidades	Tipos de habilidades que a pessoa possui
Nome da Habilidade	Nome dado à habilidade
Nível da Habilidade	Nível da habilidade que a pessoa possui
Experiência	Participações em projetos equivalentes, tarefas anteriormente realizadas similares, entre outros.
Agenda	Tempo disponível da pessoa, que pode ser destinado a tarefas de um ou mais projetos. É representado em horários e dias da semana.
Salário (Homem/Hora)	Valor recebido pela pessoa por hora de trabalho.
Modelo de Produtividade	Diferentes formas de influenciar a duração das atividades pela produtividade dos profissionais

Tabela 5.14 – Atributos de um recurso humano
Fonte: elaborada pela autora.

Na Tabela 5.15 pode-se visualizar a comparação desses elementos em cada trabalho. As fontes aparecem na ordem apresentada no parágrafo anterior. É importante salientar que, em alguns trabalhos, não estava claro todos os atributos que foram utilizados.

Elementos da Atividade	[1]	[2]	[3]	[4]	[5]	[6]
ID	N	N	N	N	S	N
Tipos de Atividades	N	N	N	S	S	N
Tipo de Habilidades	S	S	S	N	S	S
Nome da Habilidade	S	S	N	N	S	S
Nível da Habilidade	S	S	S	S	S	S
Experiência	N	N	N	N	S	N
Agenda	S	S	S	S	S	S
Salário (Homem/Hora)	N	N	S	N	S	N
Modelo de produtividade	N	N	S	N	N	N

Tabela 5.15 – Comparação de atributos de recursos humanos
Fonte: elaborada pela autora.

O modelo elaborado considerou os elementos de atividade citados na Tabela 5.14, escolhendo os mais relevantes para o trabalho em questão. Os atributos escolhidos podem ser visualizados na Tabela 5.16. Uma pessoa possui um identificador único, um nome e um salário (valor por hora). Ainda, o indivíduo possui um conjunto de competências que lhe torna capaz de executar tarefas compatíveis. Por fim, foi definido o atributo lista de desejos, que se refere a uma lista das competências que o membro deseja aprender/melhorar.

Atributos da Pessoa	Descrição
Id	Identificador da pessoa.
Nome	Nome dado à competência.
Salário	Valor recebido pela pessoa por hora de trabalho.
Competências	Conjunto de competências que o indivíduo possui (refere-se à Tabela 5.12).
Lista de desejos	Lista com as competências que o agente deseja aprender/melhorar.

Tabela 5.16 – Atributos de uma Pessoa

Fonte: elaborada pela autora.

5.5.4. Cálculo de Ganho de Experiência

A experiência adquirida por cada um dos agentes membros, a qual influencia tanto a escolha das tarefas a serem realizadas, quanto a velocidade de implementação das mesmas, foi modelada através da equação apresentada na Figura 5.11. Essa fórmula provém da área de jogos, que possui um ganho de experiência ao término de uma dada tarefa realizada dentro do jogo. Além disso, foi baseada no trabalho de Alves (2009). É importante salientar que o item da fórmula denominado “ E_{ganha} ” (experiência ganha) também é influenciado pelo conhecimento do membro, isto é, caso o membro tenha menos conhecimento que o nível de uma dada competência da atividade, o ganho de experiência será maior que uma pessoa que tenha o conhecimento de mesmo nível. Já uma pessoa com maior conhecimento que o nível da atividade ganhará menos experiência que os demais, até o ponto que não ganhe experiência nenhuma, se o seu nível for muito superior ao da tarefa.

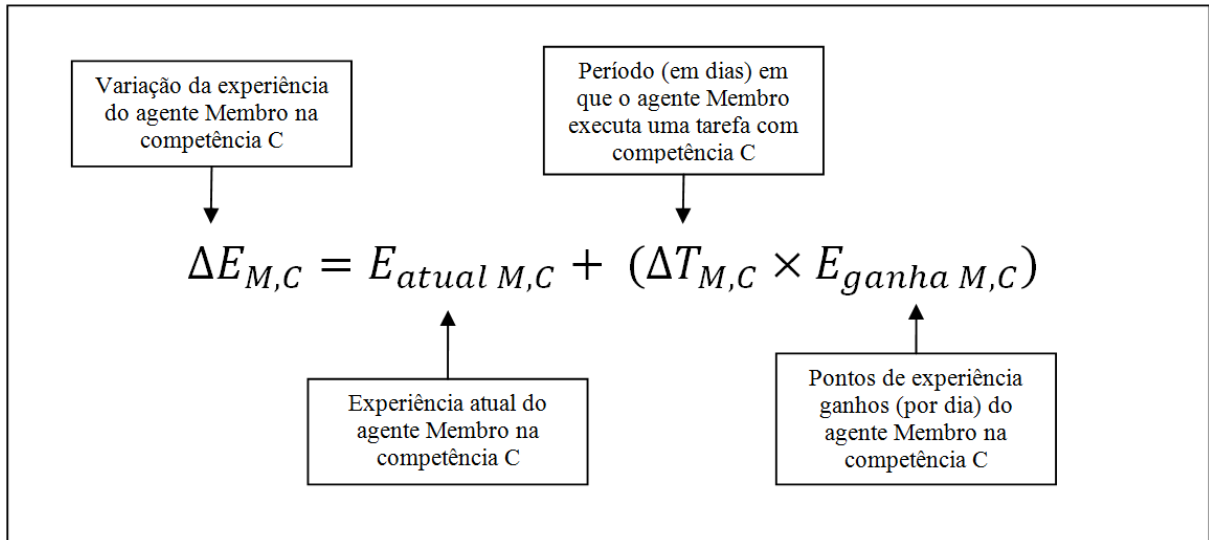


Figura 5.11 – Cálculo da Experiência Ganha pelo Agente Membro
Fonte: elaborada pela autora.

Na simulação executada neste trabalho foram utilizados os valores apresentados na Tabela 5.17 para evoluir para um nível de conhecimento. Além disso, foi considerado como pontos de experiências ganhos diário de 100. Esse valor é modificado de acordo com a diferença de níveis entre a competência de nível da atividade e do membro participante do projeto, conforme citado no parágrafo anterior. Os valores utilizados para esse cálculo são de 3 para atividades acima do nível do indivíduo e 0,2 para níveis abaixo. É importante salientar que os valores supracitados são configuráveis.

Nível	Experiência
Júnior – Pleno	26000
Pleno – Sênior	78000
Sênior – Master	130000

Tabela 5.17 – Valores de experiência para evoluir nível
Fonte: elaborada pela autora.

5.5.5. Comportamentos dos Agentes

Para atingir seus objetivos de projeto, cada agente possui certos comportamentos. Os agentes membros cooperam e competem entre si na busca por tarefas, além de tentar aprender competências que estão na sua lista de desejos. Por outro lado, o agente gestor, além de controlar a alocação de tarefas, necessita

calcular dados referentes ao projeto e gerar seu cronograma, dentro do custo e prazos pré-definidos, bem como, considerando, sempre que possível, os objetivos individuais dos membros. Os agentes identificados e seus comportamentos podem ser visualizados na Figura 5.12.

Os comportamentos dos agentes membros são:

- a) Comportamento de competição - Nesse comportamento, os agentes competem pela realização das atividades. Ele funciona da seguinte forma: quando um agente encontra uma atividade compatível, entretanto ela está pré-alocada a outro membro, o agente entra em um estado de negociação ativa, solicitando ao outro agente que lhe entregue tal tarefa.
- b) Comportamento de colaboração - De acordo com as crenças do agente (conhecimento sobre o ambiente em que está inserido), quando ele está no estado de negociação passiva (a espera do início da execução da tarefa), o agente pode colaborar com outro agente, lhe entregando uma tarefa que já estava pré-alocada e indo buscar uma nova. Essas crenças referem-se à configuração do sistema, por exemplo, se a estratégia preferida de alocação é de custo, prazo e/ou se permite aprendizagem.
- c) Comportamento de aprendizagem - Os agentes têm o intuito de aprender cada vez que executam uma nova atividade. Conforme a configuração do sistema (ambiente da simulação) é possível definir a porcentagem de tempo na qual o agente pode se dedicar à realização de tarefas compatíveis com as competências que estão na sua lista de desejos.

5.5.6. Comunicação entre Agentes

A comunicação permite que os agentes em um ambiente multiagente troquem informações que servirão de base para coordenar suas ações e realizar cooperação. A comunicação entre agentes é a troca intencional de informações causada pela produção e percepção de sinais por parte dos agentes (PESSIN, OSÓRIO, *et al.*, 2007).

Neste trabalho, a comunicação entre os agentes é realizada através da troca de mensagens. Existem dois tipos principais de comunicação. O primeiro refere-se ao gestor e o membro, onde o agente membro se comunica com o gestor para ter conhecimento da situação das atividades do projeto, bem como informar interesse

em executar dada tarefa. O segundo tipo refere-se à comunicação entre membros, onde os agentes negociam a execução de tarefas entre si.

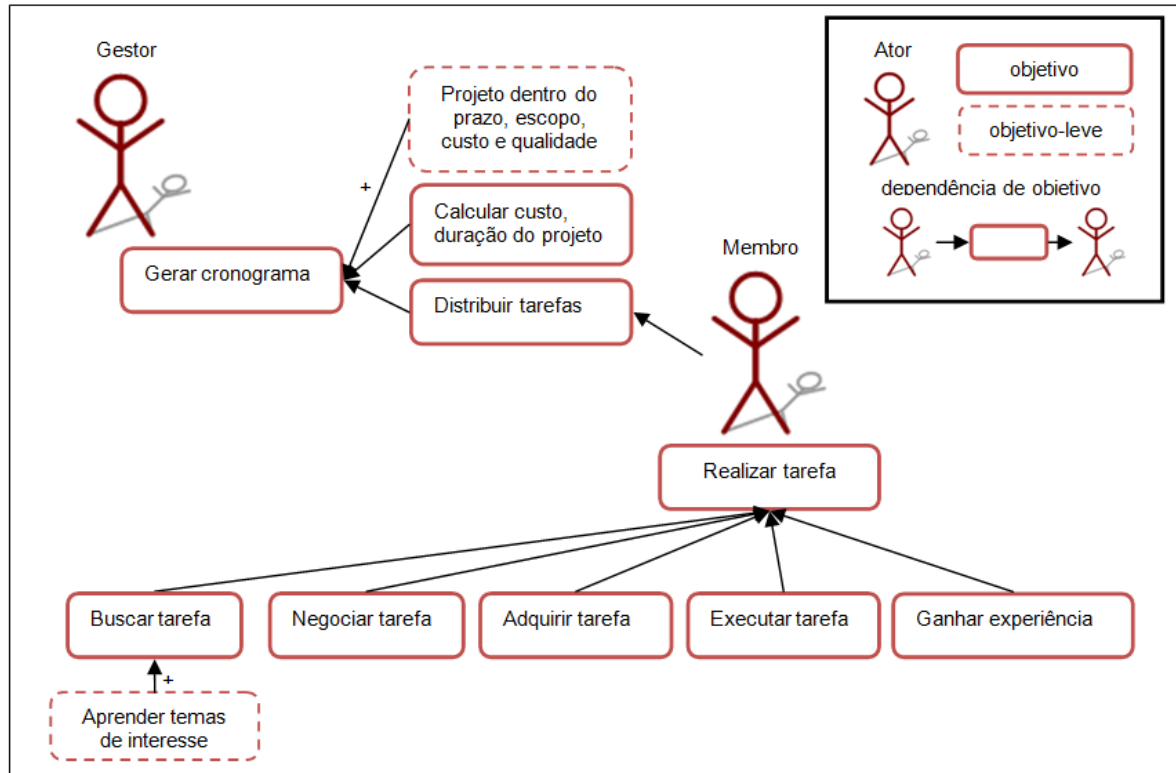


Figura 5.12 – Comportamento dos Agentes
Fonte: elaborada pela autora.

A comunicação membro-gestor pode ser visualizada na Figura 5.13. Essa figura é baseada nos diagramas de sequência apresentado na seção 5.4.4. Inicialmente, o agente membro solicita ao gestor as tarefas disponíveis do projeto e o gestor envia essa lista. Após, o membro escolhe a tarefa mais compatível (de acordo com a configuração da simulação e suas competências) e informa ao gestor o desejo de executar tal tarefa. O gestor informa ao membro se a tarefa ainda continua disponível, caso não esteja, informa o agente que pré-alocou a atividade em questão. Se a atividade continua disponível, o agente membro informa o início da execução (após todos os agentes saírem do comportamento de negociação) e o gestor atualiza o estado da tarefa como alocada. No término da atividade, o membro informa ao gestor a conclusão e o gestor atualiza o estado da tarefa como concluída. Além disso, caso não haja mais tarefas disponíveis em dado momento (por exemplo, tarefas bloqueadas a espera de conclusão de tarefas predecessoras), o agente membro espera o gestor informar a liberação de novas atividades.

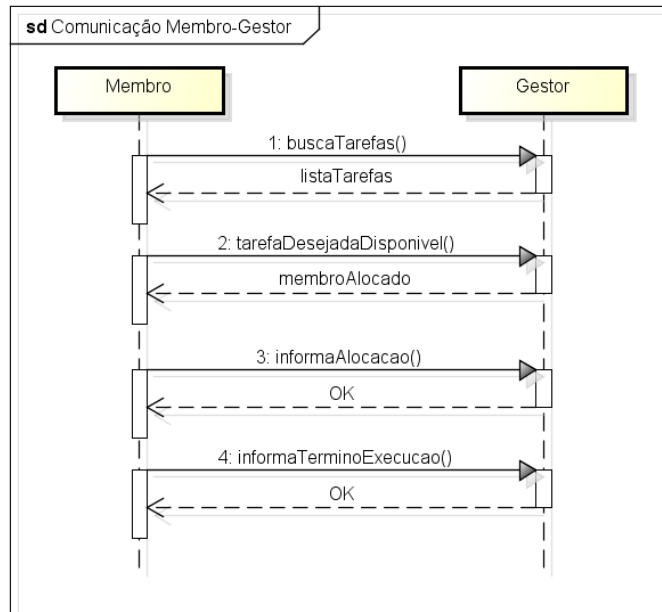


Figura 5.13 – Comunicação Membro-Gestor
Fonte: elaborada pela autora.

Na comunicação entre membros, os agentes comunicam-se entre si com o intuito de negociar a troca de atividades. Essa troca se dá quando o agente que possui a tarefa tenha menor compatibilidade com as competências dessa atividade em relação ao agente que está solicitando. Dessa forma, o primeiro entrega a tarefa ao segundo e dedica-se a buscar uma nova atividade para executar. Quando ocorre essa troca, o agente que possuía a tarefa necessita avisar ao gestor esse processo, para que o gestor possa manter o controle da alocação. A comunicação membro-membro pode ser visualizada na Figura 5.14 (baseada nos diagramas de sequência apresentados na seção 5.4.4). É importante salientar que a ação “buscaTarefa” implica em uma nova comunicação membro-gestor.

5.5.7. Configurações do Sistema

O sistema possui três formas de configuração: prazo, custo e aprendizado. A configuração por prazo visa encontrar a pessoa mais capacitada para executar uma dada tarefa. Dessa forma, o sistema busca gerar o melhor prazo possível para a entrega do projeto.

O cálculo da compatibilidade entre um membro e uma tarefa pode ser visualizado na Figura 5.15. Esse cálculo corresponde à média ponderada entre o

grau de compatibilidade (G) e o nível de competência (N) de uma tarefa. Esse cálculo permite valorizar mais uma competência com nível mais alto na tarefa. O algoritmo para calcular o grau de compatibilidade entre uma tarefa e um colaborador pode ser visualizado na Figura 5.16. O nível de competência refere-se aos níveis apresentados na Tabela 5.13 e seus valores são apresentados na Tabela 5.18.

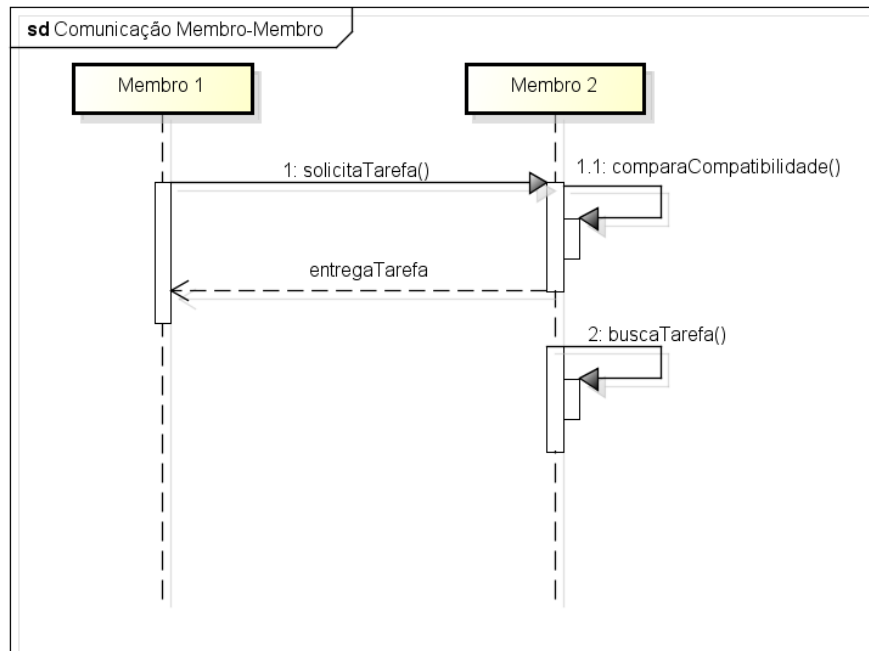


Figura 5.14 – Comunicação Membro-Membro
Fonte: elaborada pela autora.

O algoritmo apresentado na Figura 5.16 apresenta três atributos. O primeiro refere-se ao valor que se atribui a compatibilidade caso a atividade possua a mesma competência que o indivíduo. O segundo atributo refere-se ao valor de compatibilidade que se dá a uma atividade que não possui a competência idêntica a do colaborador, mas possui competências pertencentes a mesma área. O terceiro atributo refere-se à redução da compatibilidade devido à diferença de níveis técnicos, isto é, para cada nível acima ou abaixo do exigido pela tarefa é realizado um desconto de sua compatibilidade, priorizando assim habilidades do membro que possua o mesmo nível de conhecimento da atividade.

No algoritmo, o primeiro passo compara se a competência do indivíduo é a mesma da atividade. Caso seja, o coeficiente de competência recebe valor 1. Se as competências não são idênticas, compara-se a área das competências. Se ambas pertencem a mesma área, o coeficiente de competência recebe valor 0,6. Por fim, se

as competências não possuem sequer a mesma área em comum, elas são incompatíveis e o coeficiente de competência recebe o valor 0.

Além de considerar a competência em si, é importante considerar a diferença de níveis que existe entre a atividade e o indivíduo, conforme citado anteriormente, através da redução da compatibilidade. O grau de compatibilidade da fórmula da Figura 5.15 é calculado através da multiplicação entre o coeficiente de competência e o coeficiente de nível.

$$\frac{\sum_{i=1}^n G_i N_i}{\sum_{i=1}^n N_i} = \frac{G_1 N_1 + G_2 N_2 + G_3 N_3 + \dots + G_n N_n}{N_1 + N_2 + N_3 + \dots + N_n}$$

Figura 5.15 – Cálculo de Compatibilidade
Fonte: elaborada pela autora.

A configuração de sistema por custos considera um mínimo de compatibilidade que deve existir entre o indivíduo e a atividade. Nessa lista de atividades com compatibilidade mínima (valor de 0,5 na simulação atual) se escolhe a atividade que gerará menor custo para o sistema.

Por fim, a configuração por aprendizado permite a escolha da tarefa considerando os desejos dos colaboradores em aprender certas competências. A escolha da atividade se dá através da combinação entre o método de prazo (explicado anteriormente) e o método de aprendizado. Uma porcentagem será destinada à maior compatibilidade possível entre um colaborador e uma tarefa (modelo por prazo), enquanto a outra porcentagem se relaciona com o aprendizado.

O método de aprendizado calcula para cada competência de uma atividade a sua compatibilidade, escolhendo a competência que forneça maior experiência (dentro do limite diário existente). O cálculo da compatibilidade para o método do aprendizado pode ser visualizado na Figura 5.17. Conforme a equação, o grau de compatibilidade entre uma competência do membro e uma atividade é igual à competência calculada por cada método (aprendizado e prazo) multiplicado por um valor percentual que se deseja considerar na escolha.

```

MESMA_COMPETENCIA = 1
MESMA_AREA = 0,6
COEF_NIVEL_CONHECIMENTO = 0,2
//verifica competencia
se nome_comp_membro = nome_comp_tarefa
    coef_comp = MESMA_COMPETENCIA
senao se area_comp_membro = area_comp_tarefa
    coef_comp = MESMA_AREA
senao
    coef_comp = 0
//verifica niveis
dif_nivel = nivel_comp_tarefa - nivel_comp_membro
coef_niv = 1 - (dif_nivel * COEF_NIVEL_CONHECIMENTO)
Grau_Comp = coef_comp * coef_nivel

```

Figura 5.16 – Algoritmo do Grau de Compatibilidade
Fonte: elaborada pela autora.

Nível	Valor
Júnior	1
Pleno	2
Sênior	3
Master	4

Tabela 5.18 – Níveis de conhecimento de competências
Fonte: elaborada pela autora.

$$G_{C,M,T} = (C_{C,M,T} \times P_{Aprendizado}) + (C_{C,M,T} \times P_{Prazo})$$

Figura 5.17 – Fórmula de Compatibilidade do Método de Aprendizado
Fonte: elaborada pela autora.

5.6 Plataforma para Sistemas Multiagentes

Existem diversas plataformas para o desenvolvimento de sistemas multiagentes. Algumas delas podem ser visualizadas na Tabela 5.19.

Ferramenta	Desenvolvedor
AgentSheets	AgentSheets (1996)
BREVE	Spiderland (2002)

CORMAS	Cormas (2006)
JADE	Bellifemine <i>et al.</i> (2007)
JASON	Bordini <i>et al.</i> (2007)
MADKIT	MadKit (2007)
MASON	Mason (2003)
NetLogo	Wilensky (1999)
RePast	Repast (2007)
SeSAm	Sesam (2009)
SIM_AGENT	Sloman (2005)
SimCog	SimCog (2002)
SIMULA	Simula (1996)
StarLogo	StarLogo (2006)
SWARM	Swarm (2003)

Tabela 5.19 – Plataformas para desenvolvimento de sistemas multiagentes
Fonte: Adaptada de Alves (2009).

A ferramenta escolhida para o desenvolvimento do modelo apresentado de trabalho é conhecida como JADE (BELLIFEMINE, CAIRE e GREENWOOD, 2007). JADE (Java Agent Development Framework) é um framework implementado na linguagem Java e em conformidade com as especificações da FIPA (Foundation for Intelligent Physical Agents), organização IEEE de padrões para o desenvolvimento de sistemas multiagentes (FIPA, 2005).

JADE auxilia a implementação de sistemas multiagentes através de um conjunto de ferramentas gráficas que suporta depuração e fases de implantação. A plataforma de agentes pode ser distribuída através de máquinas (não importa o sistema operacional) e a configuração pode ser controlada via interface remota. A configuração pode ser alterada em tempo de execução, movendo os agentes de uma máquina a outra, quando necessário (JUN, QIANGQIANG e YINGDING, 2010).

Essa ferramenta foi escolhida para a implementação dos agentes neste trabalho pelos seguintes critérios:

- Ferramenta distribuída sem custos (software livre);
- Documentação extensa sobre seu uso, fóruns e lista de discussão ativos (SILVA, 2005);
- Linguagem de programação JAVA;
- Middleware de gerenciamento dos agentes.

6 RESULTADOS EXPERIMENTAIS

Segundo Robinson (2004, p. 53), após a construção do modelo, experimentações são realizadas na simulação para encontrar soluções para os problemas do mundo real ou obter melhor compreensão deste mundo.

Conforme mencionado no primeiro capítulo deste trabalho, o principal objetivo deste trabalho é a geração de um modelo de alocação que considere a aprendizagem das pessoas na execução de projetos buscando auxiliar os gestores de projeto no processo de tomada de decisões.

A abordagem escolhida para realizar testes sobre o modelo desenvolvido foi o uso de um exemplo de projeto de desenvolvimento de TI. Os dados de entrada utilizados foram extraídos do trabalho de Xiao *et al.* (2008) e correspondem a projetos reais. Os exemplos de usos apresentados nesse trabalho tem o intuito de demonstrar alguns dos possíveis cenários que podem ser experimentados nesse sistema. Outros cenários podem ser analisados segundo a necessidade dos gestores.

6.1 Verificação do Programa de Simulação

Conforme Davis (1992), a verificação é o processo que assegura que o modelo conceitual foi transformado com precisão adequada. Além disso, o autor afirma que a verificação consiste em duas partes. A primeira refere-se à verificação lógica e matemática, onde deve-se averiguar se os algoritmos e regras básicas funcionam como planejado e não incluem erros de lógica ou matemática. A segunda parte denominada verificação do código consiste em assegurar que as representações do modelo foram corretamente implementadas.

Ainda, Harrell *et al* (2003) declara que verificar um modelo é o mesmo que depurar a programação, buscando erros de sintaxe e semântica. Para realizar esse processo, deve-se principalmente revisar a codificação do modelo e verificar se a resposta de saída do modelo é coerente com o esperado.

Seila (1995) menciona que a verificação, embora seja um processo de depuração, pode ser complicado pois o programa de simulação pode envolver variáveis aleatórias cujos valores não podem ser previstos com antecedência.

Logo, para a verificação do programa de simulação, o mesmo foi executado passo a passo, observando se ele se comportava de acordo ao planejado e corrigindo eventuais erros encontrados.

6.2 Validação do Modelo

De acordo com Seila (1995), o processo de validação visa assegurar que o modelo computacional se aproxima adequadamente do comportamento desejado do sistema real. Ainda, conforme o autor, o processo de validação em geral envolve coleta de dados do mundo real e do sistema simulado, comparando-os e verificando se os resultados não diferem substancialmente.

Para Pereira (2000), a validação busca avaliar o quanto o modelo construído é semelhante ao sistema real que se pretendeu simular, procurando questionar se tal modelo atende ou não às finalidades de projeto, isto é, se responde convenientemente ao problema que foi levantado ao se iniciar o plano de simulação.

Para a validação do modelo apresentado neste trabalho, utilizou-se a técnica de *Black Box*, de comparação com outros modelos de simulação. Esse método busca avaliar o comportamento geral do modelo (ROBINSON, 2004). Assim, o mesmo foi dividido em duas partes, separando o modelo de melhor prazo dos demais modelos. Os resultados da simulação foram comparados com o trabalho de Xiao *et al.* (2008), encontrando valores bastante próximos.

6.3 Experimentação do Modelo

Para a experimentação do modelo, este trabalho utilizou a técnica de Análise de Variância (ANOVA). Essa técnica fornece uma forma rigorosa para identificar o efeito da mudança de fatores. Ela envolve uma série de testes de hipóteses em que é determinado se as alterações dos fatores experimentais têm efeito sobre a resposta (ROBINSON, 2004).

Conforme citado anteriormente, o modelo de simulação desenvolvido permite distintos tipos de configurações que visam verificar o funcionamento do modelo em diferentes situações. As configurações analisadas nesse trabalho podem ser visualizadas na Tabela 6.1.

Configuração	Descrição
Melhor prazo	Seleciona os membros da equipe que possuam as competências mais adequadas para realizar uma dada tarefa.
Melhor custo	Respeitando uma certa compatibilidade entre as competências da tarefa e do membro da equipe, seleciona o indivíduo de menor salário por hora.
Aprendizado 30%	Balanceamento 30% aprendizado e 70% melhor prazo.
Aprendizado 100%	Respeitando uma compatibilidade mínima em relação às competências, seleciona a pessoa com maior compatibilidade em relação ao aprendizado.

Tabela 6.1 – Configurações do Modelo
Fonte: elaborada pela autora.

Conforme mencionado no início do capítulo, os dados de entrada utilizados foram retirados do trabalho de Xiao *et al.* (2008) e podem ser visualizados no ANEXO A – Dados DO PROJETO, ANEXO B – Dados DOS RECURSOS HUMANOS e APÊNDICE A – DADOS DE APRENDIZAGEM.

As tabelas apresentadas na próxima seção mostram os resultados da simulação para cada uma das configurações citadas acima. Maiores detalhes com respeito a alocação das atividades podem ser encontrados no APÊNDICE B – PLANEJAMENTO MELHOR PRAZO, APÊNDICE C – PLANEJAMENTO MELHOR CUSTO, APÊNDICE D – PLANEJAMENTO APRENDIZADO 30% E APÊNDICE E – PLANEJAMENTO APRENDIZADO 100%.

6.4 Resultados Encontrados

Replicação	Custo	Experiência	Dias úteis
1	51640	18400	96
2	53000	18400	96
3	51640	18400	96
4	51640	18400	96
5	53360	18400	96
6	51720	18400	96
7	51640	18400	96
8	51640	18400	96
9	53000	18400	96
10	51720	18400	96

Média	\$ 52.100,00	18400	96
Desvio Padrão	\$ 711,3680	0	0

Tabela 6.2 – Configuração "Melhor Prazo"
Fonte: Elaborada pela autora.

Replicação	Custo	Experiência	Dias úteis
1	44640	16200	98
2	44640	16200	98
3	44640	16200	98
4	44640	16200	98
5	45120	16560	101
6	44640	16200	98
7	44640	16200	98
8	44640	16200	98
9	44640	16200	98
10	44640	16200	98
Média	\$ 44.688,00	16236	98,3
Desvio Padrão	\$ 151,7893	113,841996	0,9486833

Tabela 6.3 – Configuração "Melhor Custo"
Fonte: Elaborada pelo autora.

Replicação	Custo	Experiência	Dias úteis
1	52520	36100	106
2	53160	25900	98
3	52520	36100	106
4	49960	25900	98
5	54160	36100	106
6	53160	25900	98
7	52520	36100	106
8	54160	36100	106
9	52520	36100	106
10	52520	36100	106
Média	\$ 52.720,00	33040	103,6
Desvio Padrão	\$ 1.171,514	4927,06809	3,86436713

Tabela 6.4 – Configuração "Aprendizado 30%"
Fonte: Elaborada pela autora.

Replicação	Custo	Experiência	Dias úteis
1	62400	44920	110
2	61800	46340	113
3	61280	44920	110
4	55360	51140	127
5	56480	51140	127

6	61920	50020	110
7	56000	45580	110
8	61800	46340	113
9	61800	46340	113
10	56000	45580	110
Média	\$ 59.484,00	47232	114,3
Desvio Padrão	\$ 3.056,182	2513,38108	6,83211371

Tabela 6.5 – Configuração "Aprendizado 100%"

Fonte: Elaborada pela autora.

Para verificarmos a significância estatística dos experimentos, utilizamos a Análise de Variância (ANOVA) de dois fatores com replicação e alfa igual a 0,05 (LES, 2002). As próximas tabelas apresentam os resultados do teste ANOVA.

Melhor Prazo	Custo	Experiência	Dias úteis	Total
Contagem	10	10	10	30
Soma	521000	184000	960	705960
Média	52100	18400	96	23532
Variância	506044,4444	0	0	480058477,2

Tabela 6.6 – Resumo Melhor Prazo

Fonte: Elaborada pela autora.

Melhor Custo	Custo	Experiência	Dias úteis	Total
Contagem	10	10	10	30
Soma	446880	162360	983	610223
Média	44688	16236	98,3	20340,76667
Variância	23040	12960	0,9	351526461,4

Tabela 6.7 – Resumo Melhor Custo

Fonte: Elaborada pela autora.

Aprendizado 30%	Custo	Experiência	Dias úteis	Total
Contagem	10	10	10	30
Soma	527200	330400	1036	858636
Média	52720	33040	103,6	28621,2
Variância	1372444,444	24276000	14,93333333	495384509,7

Tabela 6.8 – Resumo Aprendizado 30%

Fonte: Elaborada pela autora.

Aprendizado 100%	Custo	Experiência	Dias úteis	Total
Contagem	10	10	10	30
Soma	594840	472320	1143	1068303
Média	59484	47232	114,3	35610,1
Variância	9340248,889	6317084,444	46,67777778	682439696,7

Tabela 6.9 – Resumo Aprendizado 100%
Fonte: Elaborada pela autora.

Total	Custo	Experiência	Dias úteis
Contagem	40	40	40
Soma	2089920	1149080	4122
Média	52248	28727	103,05
Variância	30737329,23	166988319	65,48461538

Tabela 6.10 – Resumo Total
Fonte: Elaborada pela autora.

Fonte de Variação	SS ²	df ³	MS ⁴	F	P-valor	F crit
Amostra	3993954497	3	1331318166	381,7592715	2,58689E-57	2,688691468
Colunas	54555516868	2	27277758434	7821,974833	1,4087E-117	3,080386863
Interação	3340717374	6	556786229	159,6600352	2,53153E-51	2,183656883
Dentro	376630962,6	108	3487323,728			
Total	62266819702	119				

Tabela 6.11 – Resultados ANOVA
Fonte: Elaborada pela autora.

A hipótese nula considera que a média dos fatores são iguais. Com o resultado da ANOVA pode-se verificar que os valores F (variância dos fatores) são todos superiores ao $F_{\text{crítico}}$. Além disso, o P-valor é inferior a alfa (0,05), mostrando que os resultados são estatisticamente significativos. Isso significa que existe uma diferença entre as configurações para a empresa. Portanto, rejeita-se a hipótese nula.

A partir dessa constatação, analisa-se os resultados em relação a cada um dos fatores. Na Figura 6.1 pode-se visualizar a comparação do custo médio para cada tipo de configuração. Conforme o esperado, em Aprendizado 100% o custo médio é superior aos demais (\$ 59.484,00), ou seja, 14,17% maior que o modelo geralmente adotado pelas empresas (Melhor Prazo). Isso ocorre pois os membros da equipe menos qualificados demoram mais tempo para realizar as atividades, ocasionando maior custo. Ainda, obviamente, em Melhor Custo foi encontrado o menor custo médio (\$ 44.688,00), 14,23% menor que o modelo Melhor Prazo. Além disso, observa-se que a diferença entre os custos médios das configurações Melhor Prazo (\$ 52.100,00) e Aprendizado 30% (\$ 52.720,00) pode ser considerada

² SS: Soma dos quadrados.

³ df: Graus de Liberdade.

⁴ MF: Média quadrada.

pequena (1,19%) e permite aos membros da equipe (na realização de tarefas compatíveis com seus desejos de aprendizado) maior motivação e ganho de experiência. A diferença de porcentagens entre os custos (comparando com o modelo Melhor Prazo, geralmente adotado pelas empresas) é apresentada na Tabela 6.12.

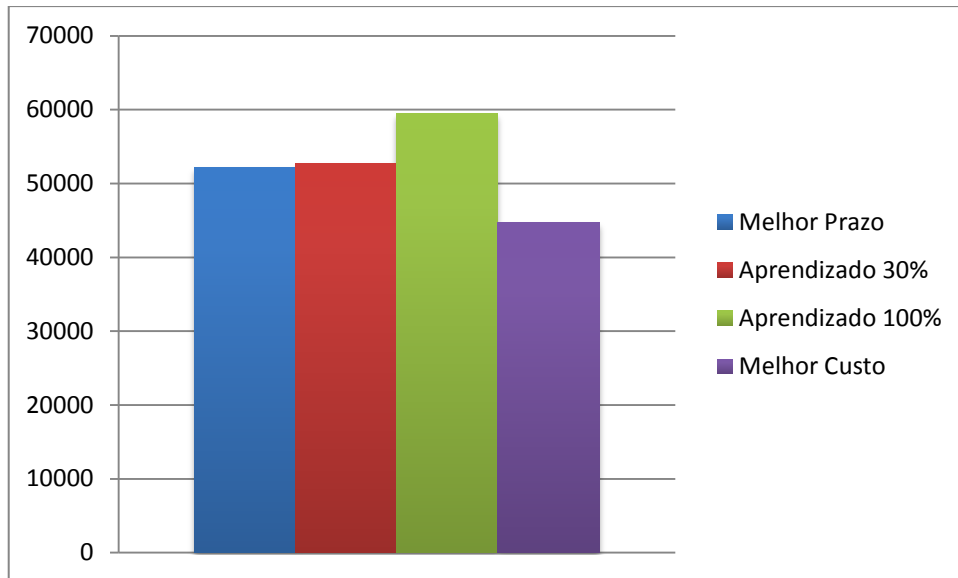


Figura 6.1 – Comparação entre custos médios
Fonte: Elaborada pela autora.

Melhor Prazo	\$ 52.100,00	100,000%
Aprendizado 30%	\$ 52.720,00	101,190%
Aprendizado 100%	\$ 59.484,00	114,173%
Melhor Custo	\$ 44.688,00	85,774%

Tabela 6.12 – Comparação de custos médios em porcentagem
Fonte: Elaborada pela autora.

Na Figura 6.2 pode-se visualizar a comparação dos prazos médios de execução do projeto. No modelo Melhor Prazo encontrou-se o menor número de dias úteis (96). Os dias úteis das demais configurações aumentam de forma progressiva, onde o maior prazo médio é encontrado no modelo Aprendizado 100% (aproximadamente 115 dias úteis e um aumento de 19,1%). A diferença entre os modelos Melhor Prazo e Aprendizado de 30% é de aproximadamente 8%, valor maior que o apresentado em custos, embora se visualizar o projeto como um todo, seria de aproximadamente 8 dias úteis de atraso, valor aceitável para uma empresa.

As porcentagens de cada configuração comparadas com o modelo Melhor Prazo podem ser visualizadas na Tabela 6.13.

Na Figura 6.3 pode-se visualizar a comparação das experiências ganhas em cada uma das configurações. Como esperado, o modelo Aprendizado 100% possui o maior ganho de experiência médio (47232 pontos). O aumento em relação ao modelo geralmente adotado pela empresa foi de aproximadamente 157%. O modelo balanceado (Aprendizado 30%) obteve um ganho de experiência em comparação ao modelo de Melhor Prazo de aproximadamente 179,6%. O modelo de melhor custo teve um decréscimo de experiência em relação ao modelo base. Isso se explica pelo fato que, na simulação dos dados em questão, membros da equipe com maior nível de conhecimento em relação a uma dada atividade e salários inferiores foram assignados a tal tarefa, ganhando menos pontos de experiência nessa situação. Para melhor entendimento dessa questão, sugere-se a análise do APÊNDICE C – PLANEJAMENTO MELHOR CUSTO. Os valores diferenças em porcentagens entre cada tipo de configuração são apresentados na Tabela 6.14.

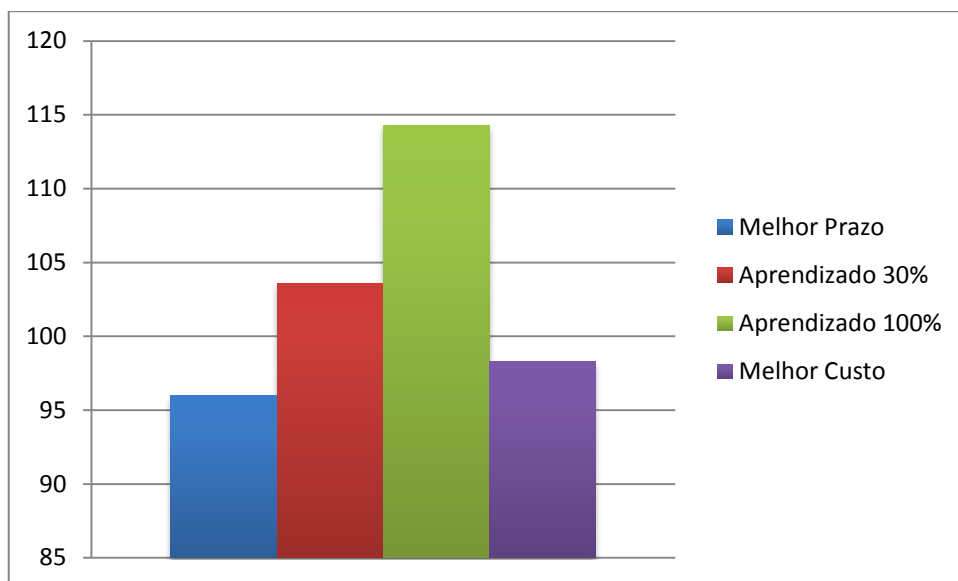


Figura 6.2 – Comparação entre prazos médios (dias úteis)
Fonte: Elaborada pela autora.

Melhor Prazo	96	100,000%
Aprendizado 30%	103,6	107,917%
Aprendizado 100%	114,3	119,063%
Melhor Custo	98,3	102,396%

Tabela 6.13 – Comparação de prazos médios (dias úteis) em porcentagem
Fonte: Elaborada pela autora.

Na Figura 6.4 é apresentado um resumo das configurações. É possível observar que o ganho de experiência nos métodos de aprendizado é consideravelmente superior ao modelo Melhor Prazo. Embora os custos e prazos cresçam, os mesmos aumentam em uma proporção inferior ao ganho de experiência.

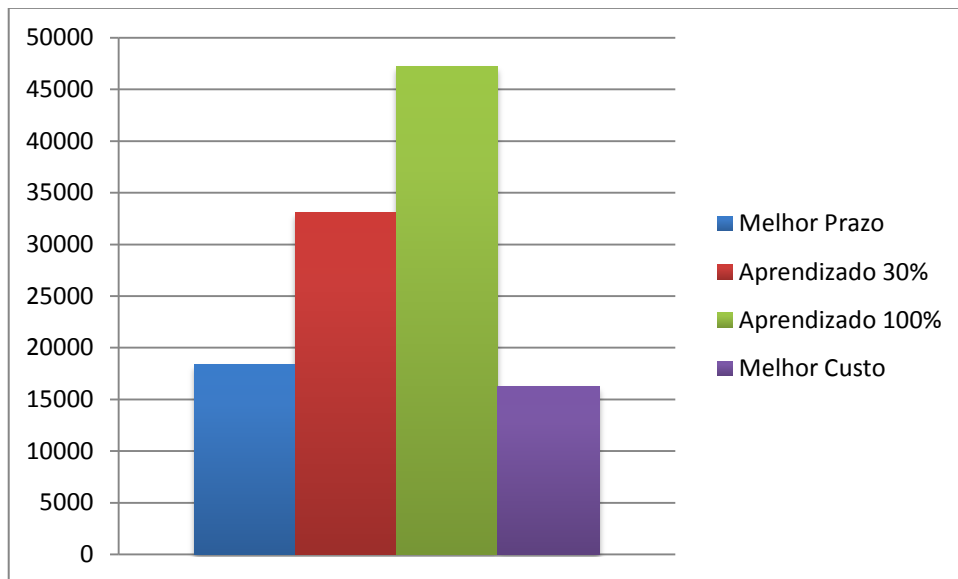


Figura 6.3 – Comparação entre pontos de experiência
Fonte: Elaborada pela autora.

Melhor Prazo	18400	100,000%
Aprendizado 30%	33040	179,565%
Aprendizado 100%	47232	256,696%
Melhor Custo	16236	88,239%

Tabela 6.14 – Comparação entre pontos de experiência em porcentagem
Fonte: Elaborada pela autora.

Segundo os resultados apresentados, pode-se considerar que o modelo misto (Aprendizado 30%), embora tenha um maior custo e prazo de projeto, o prejuízo para o negócio é menor que o ganho de experiência obtido com tal modelo, pois além da empresa permitir que seus funcionários tenham participação na escolha das atividades, essas pessoas estarão mais motivadas e desenvolverão suas competências mais rapidamente, trazendo benefícios a longo prazo para empresa, com a execução de atividade de forma mais ágil futuramente por esses empregados.

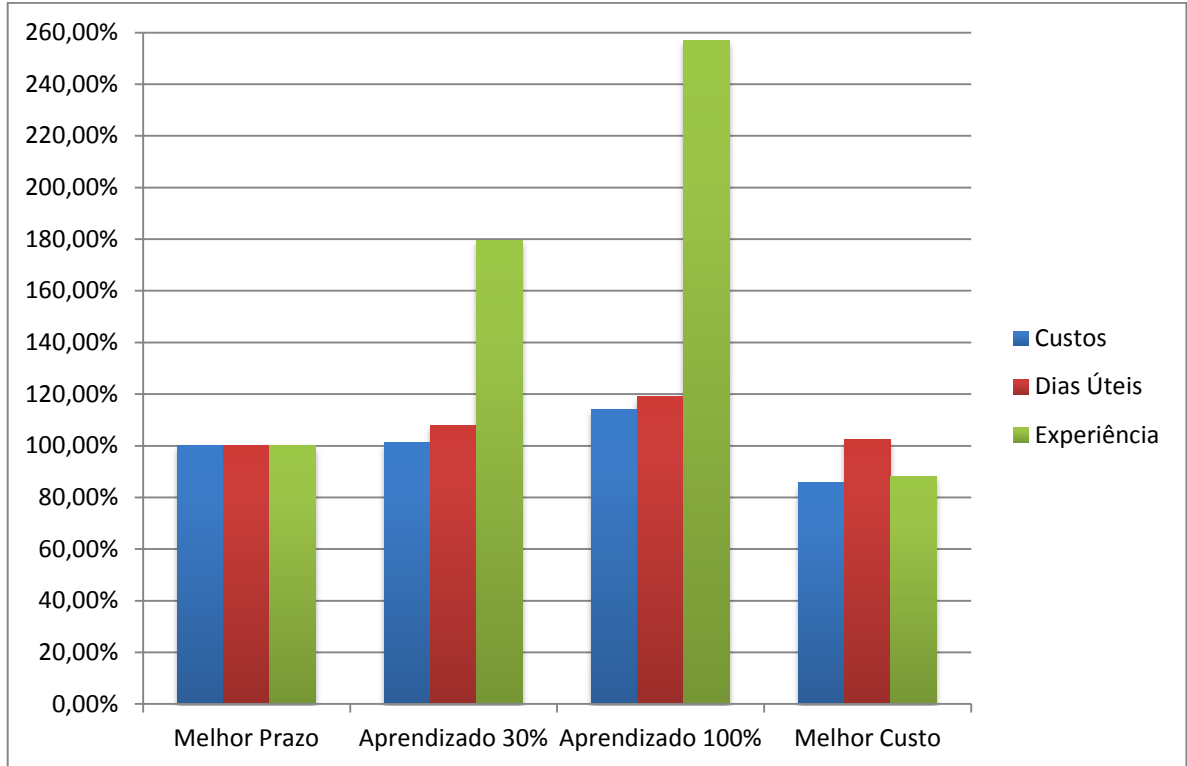


Figura 6.4 – Resumo das configurações
Fonte: Elaborada pela autora.

7 CONSIDERAÇÕES FINAIS

Este trabalho propõe um modelo de alocação de tarefas aos participantes de uma equipe de projetos de software que considere, além dos objetivos organizacionais, os individuais dos colaboradores. Através desse modelo, os gestores de projeto podem tomar decisões com maior segurança, visando obter maior desempenho na realização dos projetos de software e a melhoria de qualificação dos membros participantes.

Para a geração do modelo em questão, primeiramente foram estudados referenciais da área de recursos humanos e gestão de projetos, bem como, modelos equivalentes propostos na literatura. Após, estudou-se sistemas multiagentes e suas modelagens para a construção do modelo em questão. Assim, foi possível elaborar um modelo conceitual que considerasse os anseios profissionais dos trabalhadores, através de agentes de software.

Esse modelo prevê vários tipos de configuração que podem ser analisados pelos gerentes. Uma das configurações, denominada Melhor Prazo e geralmente adotada pelas organizações, busca a melhor compatibilidade entre as competências de uma pessoa e uma dada atividade. Nesse modelo gera-se um planejamento da execução do projeto que obtenha o melhor prazo de entrega possível. Outro cenário possível é a geração de um plano de execução que obtenha o menor custo. Além disso é possível considerar um certo nível que a empresa permite aos seus colaboradores escolher tarefas que tenham alguma competência que eles desejam desenvolver.

A simulação foi implementada em linguagem Java, com apoio do framework JADE (BELLIFEMINE, CAIRE e GREENWOOD, 2007), que auxilia no gerenciamento dos comportamentos dos agentes, bem como, na comunicação entre eles, abstraindo essa preocupação do programador.

Para validação da simulação, o modelo proposto foi comparado com o modelo de Xiao *et al.* (2008) e testado através do formato de experimentação estatística ANOVA (LES, 2002), mostrando a significância dos dados de saída.

Os resultados geraram os valores de prazo (dias úteis), custo e pontos de experiência para cada uma das configurações citadas anteriormente. Através da comparação com o modelo Melhor Prazo, foi possível perceber que, embora os modelos de aprendizado aumentassem o custo e prazo do projeto, o ganho de

experiência era relativamente maior. Na opinião da autora, segundo os resultados encontrados, os benefícios que a empresa ganha, com o desenvolvimento de seus colaboradores e aumento de sua motivação, são maiores que o pequeno atraso do projeto e aumento do custo apresentado.

Em relação ao uso de agentes e a modelagem utilizada, considera-se que este tipo de abordagem facilitou o desenvolvimento do trabalho, já que agentes são bastante adequados para simular modelos baseados em seres humanos.

É importante mencionar como contribuição deste trabalho, a publicação deste no evento ENEGEP (Encontro Nacional de Engenharia de Produção) 2011.

As próximas seções apresentam as limitações do estudo, trabalhos correlatos e trabalhos futuros.

7.1 Limitações do Estudo

Este modelo tem como principal enfoque a alocação de recursos humanos em atividades de projetos de software. Não faz parte do escopo deste trabalho a alocação de recursos humanos em outros tipos de projeto, bem como a alocação de outros quaisquer outros tipos de recursos.

Uma limitação é falta de aplicação em projetos reais atuais em empresas de software. Embora parte do modelo foi testada com dados publicados, a configuração de aprendizado por falta de tempo não pôde ser testada em organizações. Considera-se importante a experimentação dos dados de aprendizado, acompanhando o desenvolvimento dos membros da equipe de projeto.

Outra limitação do trabalho é a escolha dos valores de ganho de experiência. É importante salientar que, esse valor, embora arbitrário, foi considerado analisando o tempo que um profissional leva para evoluir de um nível de aprendizado a outro. Como trabalhos futuros, sugere-se maior estudo e análise com respeito ao ganho de aprendizado do membro com o decorrer do tempo.

Por fim, acredita-se que o modelo deveria ser testado mais cenários, tais como projetos maiores ou diversidade de projetos. Embora a execução de validação tenha sido realizada com mais de um projeto – comparação com o modelo de Xiao *et al.* (2008) – os resultados encontrados com as configurações apresentadas foram executadas em apenas um projeto.

7.2 Trabalhos Correlatos

Esta dissertação apresenta dois aspectos que podem ser comparados a outros trabalhos:

- Alocação de pessoas a tarefas em projetos;
- Simulação multiagentes.

Fatima *et al.* (2000) e Fatima e Wooldridge (2001) apresentam uma política de adaptação organizacional para sistemas multiagentes denominada TRACE. Essa proposta permite que uma coleção de organizações multiagentes possam atribuir dinamicamente tarefas e recursos entre elas.

Schnaider (2003) apresenta uma abordagem para planejamento de alocação de recursos humanos em projetos de desenvolvimento de software fundamentada nos conceitos de gerência de conhecimento e ambientes de desenvolvimento de software orientados à organização (ADSOrg) e que possui como pilar a experiência pessoal do gerente de projeto.

Barreto (2005), propõe uma ferramenta de apoio à decisão no contexto da alocação de recursos humanos. Essa ferramenta foi construída utilizando a teoria das restrições.

Ainda, Jinghua e Wenjian (2005) apresentam um *framework* de planejamento de alocação de recursos em multi-projetos utilizando abordagem distribuída de multiagentes em empresas aeroespaciais.

Alba e Chicano (2007) propõem uma ferramenta automatizada de alocação de pessoas a tarefas de projetos utilizando algoritmos genéticos. Essa ferramenta testa diferentes cenários de projetos de software, levando em conta as variáveis de custo e duração do projeto. Ainda, descreve recursos humanos usando atributos como habilidades, salários e grau de dedicação ao projeto.

Xiao *et al.* (2008) apresentam um método para alocação de recursos humanos dirigido a restrições para o processo de desenvolvimento e manutenção de software em organizações maduras de software, como as que possuem CMMI⁵ nível 4 ou maior. Essa proposta considera as diferenças de carga de trabalho e custo se recursos humanos de diferentes capacidades executam a mesma atividade.

⁵ CMMI: abordagem de melhoria de processos utilizada em desenvolvimento de software (SOFTWARE ENGINEERING INSTITUTE, 2011).

Wang, Gong e Yan (2009) apresentam um método de alocação de recursos humanos em projetos baseado em um algoritmo de otimização por enxame de partículas (Particle Swarm Optimization – PSO). Chang (2010) propõe um método de alocação de recursos humanos em empresas baseado em redes neurais. Já Weng *et al.* (2010) propõem um método de alocação de recursos humanos em multi-projetos utilizando o modelo de análise envoltória de dados (DEA). Ainda, Daojin (2010) propõe um método de alocação de recursos humanos baseado no processo de hierarquia analítica e matemática Fuzzy.

Entretanto, nenhum deles apresenta a possibilidade de considerar cenários para desenvolver os recursos humanos da organização, através da realização de tarefas apropriadas, como pode ser visualizado na Tabela 7.1. Ainda, conforme Dong *et al.* (2008) e Weng *et al.* (2010) há pouco estudo relacionado a multi-projetos, característica contemplada neste trabalho.

Trabalho	Multi-Projeto	Custo	Negociação	Método	Aprendizado
Fatima <i>et al.</i> (2000)	Sim	Sim	Sim	Sistemas Multiagentes	Não
Schneider (2003)	Não	Sim	Não	ADSOrg	Não
Barreto (2005)	Não	Sim	Não	Dirigido a restrições	Não
Jinghua e Wenjian (2005)	Sim	Indeterminado	Sim	Sistemas Multiagentes	Não
Alba e Chicano (2007)	Indeterminado	Sim	Não	Algoritmos Genéticos	Não
Xiao <i>et al.</i> (2008)	Sim	Sim	Não	Dirigido a Restrições	Não
Wang <i>et al.</i> (2009)	Indeterminado	Sim	Não	Otimização por enxame de partículas	Não
Weng <i>et al.</i> (2010)	Sim	Sim	Não	Análise envoltória de dados	Não
Chang (2010)	Indeterminado	Sim	Não	Redes Neurais	Não

Daojin (2010)	Indeterminado	Sim	Não	Fuzzy	Não
Esta dissertação	Sim	Sim	Sim	Sistemas Multiagentes	Sim

Tabela 7.1 - Comparação entre trabalhos correlatos e essa dissertação
 Fonte: Elaborado pelo autor.

7.3 Trabalhos Futuros

Essa pesquisa levanta a possibilidade de outras pesquisas possíveis para que o modelo em questão seja evoluído, a saber:

- Adaptar o modelo para outros tipos de projetos fora do universo de Tecnologia, considerando modelos de competências diferentes;
- Comparar o modelo apresentado com outros propostos na literatura;
- Aplicação prática do modelo em projetos de software, acompanhando e mensurando o desenvolvimento e aprendizado dos membros da equipe, bem como, quanto a escolha das atividades afeta na motivação dos mesmos;
- Desenvolver uma interface gráfica para adicionar atividades, recursos humanos e gerenciar as configurações do modelo;
- Geração do gráfico GANTT de forma automática;
- Adição de novas configurações, conforme as necessidades de empresa;
- Adaptar o modelo para que mais de uma pessoa possa executar uma tarefa ao mesmo tempo;
- Considerar a alocação de outros tipos de recursos no modelo;
- Adição de medidas de avaliação de desempenho dos recursos humanos propostas na literatura para definir quais competências o profissional precisa desenvolver;
- Executar o modelo em cenários mais complexos, projetos maiores, ambientes de multi-projetos;
- Adaptar o modelo para aceitar métodos de desenvolvimento de software, como métodos ágeis, por exemplo.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- ABREU, M. C. C. **Analisando o desempenho do ClassAge: Um Sistema Multiagentes para Classificação de Padrões**. Universidade Federal do Rio Grande do Norte. Natal, p. 150. Dissertação (Mestrado em Sistemas e Computação). 2006.
- ACUÑA, S. T.; JURISTO, N.; MORENO, A. M. Emphasizing Human Capabilities in Software Development. **IEEE Software**, v. 23, n. 2, p. 94-101, Mar/Abr 2006.
- AGENTSHEETS, INC. AgentSheets, 1996. Disponível em: <<http://www.agentsheets.com/>>. Acesso em: 17 jul. 2011.
- ALBA, E.; CHICANO, F. Software project management with GAs. **Journal Information Sciences**, New York, USA, v. 177, n. 11, p. 2380-2401, Jun. 2007.
- ALVES, F. O. **SIMULAÇÃO MULTI-AGENTE EM GESTÃO DE PROJETOS DE SOFTWARE EM AMBIENTES DE PROGRAMAÇÃO EXTREMA**. Universidade Federal do Rio Grande do Sul. Porto Alegre, p. 127. Dissertação (Mestrado em Administração). 2009.
- ANTONIOL, G. et al. Assessing staffing needs for a software maintenance project through queuing simulation. **IEEE Transactions on Software Engineering**, v. 30, p. 43-58, 2004.
- ARGONNE NATIONAL LABORATORY. Repast Suite. **RePast**, 2007. Disponível em: <<http://repast.sourceforge.net/>>. Acesso em: 30 jul 2011.
- ATKINSON, R. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. **International Journal of Project Management**, v. 17, n. 6, p. 337-342, Dec. 1999.
- BAE, D.-H. Software Process Improvement for Small Organizations. **31st Annual International Computer Software and Applications Conference**, Beijing, v. 1, p. 17, Jul. 2007.
- BARBAZETTE, J. **The trainer's journey to competence: tools, assessments, and models**. San Francisco: John Wiley and Sons, 2005.
- BARRETO, A. S. **APOIO À ALOCAÇÃO DE RECURSOS HUMANOS EM PROJETOS DE SOFTWARE: UMA ABORDAGEM BASEADA EM SATISFAÇÃO DE RESTRIÇÕES**. Universidade Federal do Rio de Janeiro. Rio de Janeiro, p. 104. Dissertação (Mestrado em Engenharia de Sistemas e Computação). 2005.
- BELLIFEMINE, F.; CAIRE, G.; GREENWOOD, D. **Developing multi-agents systems with Jade**. England: John Wiley & Sons Ltd., 2007.
- BHATTI, M. W. et al. **A methodology to manage the changing requirements of a software project**. International Conference on Computer Information Systems and Industrial Management Applications (CISIM). [S.l.]: [s.n.]. 2010. p. 319-322.

- BONABEAU, E. Agent-based modeling: methods and techniques for simulating human systems. **Proceeding of the Nacional Academy of Sciences**, v. 99, n. 3, p. 7280-7287, May 2002.
- BORDINI, R. H.; HÜBNER, J. F.; WOOLDRIDGE, M. **Programming Multi-Agent Systems in AgentSpeak using Jason**. [S.l.]: Wiley-Interscience, 2007.
- BRATMAN, M. **Intention, Plans and Practical Reason**. Cambridge, Massachusetts: Harvard University Press, 1987.
- BRAUBACH, L.; POKAHR, A.; LAMERSDOF, W. Jadex: A BDI Agent System Combining Middleware and Reasoning. In: CALISTI, M. A. W. M. A. B. S. A. H. M. A. U. R. A. C. M. A. K. M. **Software Agent-Based Applications, Platforms and Development Kits**. [S.l.]: Birkhäuser Basel, 2005. p. 143-168.
- BRENNER, W.; ZARNEKOW, R.; WITTING, H. **Intelligent Software Agents: Foundations and Applications**. [S.l.]: Springer-Verlag, 1998.
- BRESCIANI, P. et al. Modeling Early Requirements in Tropos: A Transformation Based Approach. In: WOOLDRIDGE, M. J.; WEIß, G.; CIANCARINI, P. **Agent-Oriented Software Engineering II**. Montreal: Springer-Verlag, v. 2222, 2002. p. 151--168.
- CARVALHO, T. C. Y. **DESENVOLVIMENTO DE UM JOGO DE EMPRESAS PARA SER UTILIZADO NO ENSINO DE GESTÃO DE PROJETOS**. Escola Politécnica da Universidade de São Paulo. São Paulo, p. 101. Monografia (Engenharia de Produção). 2004.
- CAUDRON, S. O que motiva os empregados. **HSM Management**, São Paulo, n. 1, p. 82-86, Mar/Abr 1997.
- CELAYA, J. R.; DESROCHERS, A. A.; GRAVES, R. J. **Modeling and Analysis of Multi-agent Systems using Petri nets**. IEEE International Conference on Systems, Man and Cybernetics. Montreal: [s.n.]. 2007. p. 1439-1444.
- CERVENKA, R.; TRENCANSKY, I. **AML: The Agent Modeling Language**. Berlin: Birkäuser, 2000.
- CHANG, C. K. et al. Software Project Management Net: A New Methodology on Software Management. **The Twenty-Second Annual International Computer Software and Applications Conference**, Vienna, 19 Aug. 1998. 534.
- CHANG, N. **The Application of Neural Network to the Allocation of Enterprise Human Resources**. 2nd International Conference on e-Business and Information System Security (EBISS). Wuhan: [s.n.]. 2010. p. 1.
- CHIAVENATO, I. **Introdução à Teoria Geral da Administração**. São Paulo: Makron Books, 1997.

- CHIAVENATO, I. **Recursos Humanos: O Capital Humano das Organizações**. São Paulo: Editora Atlas, 2004.
- CHIAVENATO, I. **Gestão de Pessoas: o novo papel dos recursos humanos nas organizações**. 3. ed. Rio de Janeiro: Elsevier, 2010.
- CISCON, L. A. **UM ESTUDO E UMA FERRAMENTA DE GERÊNCIA DE PROJETOS COM DESENVOLVIMENTO ÁGIL DE SOFTWARE**. Universidade Federal de Minas Gerais. Belo Horizonte, p. 164. Dissertação (Mestrado em Ciência da Computação). 2009.
- COELHO, A. G. N. **Apoio à Gerência de Recursos em ODE**. Universidade Federal do Espírito Santo. Vitória, p. 169. Monografia (Ciência da Computação). 2007.
- COLARES, J. C. A Contribuição Behaviorista Para A Administração, 2008. Disponível em: <<http://www.webartigos.com/articles/4340/1/A-Contribuicao-Behaviorista-Para-A-Administracao/pagina1.html>>. Acesso em: 20 Mar. 2011.
- CORMAS. Natural Resources and Multi-Agent Simulations, 2006. Disponível em: <<http://cormas.cirad.fr/indexeng.htm>>. Acesso em: 17 jul. 2001.
- CRESWELL, J. W. **Research design: qualitative, quantitative and mixed method approaches**. California: Sage Publications, 2003.
- CZARNOWSKI, I.; JEDRZEJOWICZ, P. Implementation and Performance Evaluation of the Agent-Based Algorithm for ANN Training. **Int. J. Know.-Based Intell. Eng. Syst.**, Amsterdam, v. 14, n. 1, p. 1-10, Jan. 2010.
- DAOJIN, F. **Research on the comprehensive evaluation of the human resource allocation based on Analytic Hierarchy Process and Fuzzy Mathematics**. 2nd International Conference on Industrial and Information Systems (IIS). Dalian: [s.n.]. 2010. p. 491-494.
- DÁRIO, C. F. B. **Uma Metodologia Unificada para o Desenvolvimento de**. Universidade Estadual de Campinas. Campinas, p. 200. Dissertação (Mestrado em Engenharia Elétrica). 2005.
- DAVIS, P. K. **Generalizing concepts of verification, validation and accreditation (VV&A) for Military Simulations**. Tech. Rep. R-4249-ACQ, The RAND Corporation. Santa Monica, CA. 1992.
- DENNETT. **The Intentional Stance**. [S.l.]: Bradford Books, 1987.
- DINGSOYR, T.; ROYRVIK, E. Skills Management as Knowledge Technology in a Software Consultancy Company. In: _____ **Advances in Learning Software Organisations**. [S.l.]: Springer-Verlag, 2001. p. 96-107.
- DONG, F. et al. Software Multi-project Resource Scheduling: A Comparative Analysis. In: WANG, Q.; WANG, P. D.; RAFFO, D. M. **Making Globally**

- Distributed Software Development a Success Story.** [S.l.]: [s.n.], v. 5007, 2008. p. 63-75.
- DUARTE, R. N. **Simulação computacional: análise de uma célula de manufatura em lotes do setor de autopeças.** Universidade Federal de Itajubá. Itajubá, p. 168. Dissertação (Mestrado em Engenharia da Produção). 2003.
- DURAI, P. **Human Resource Management.** Singapore: Pearson Education, 2010.
- DUTRA, J. S. **Competências: conceitos e instrumentos para gestão de pessoas na empresa.** São Paulo: Gente, 2004.
- FATIMA, S. S.; UMA, G.; TOLETY, S. P. **TRACE - an adaptive organizational policy for multi agent systems.** Fourth International Conference on MultiAgent Systems. Boston, MA: [s.n.]. 2000. p. 383.
- FATIMA, S. S.; WOOLDRIDGE, M. **Adaptive task resources allocation in multi-agent systems.** Fifth International Conference on Autonomous Agents. [S.l.]: [s.n.]. 2001. p. 537-544.
- FIPA. Foundation for Intelligent Physical Agents, 2005. Disponível em: <<http://www.fipa.org/>>. Acesso em: 30 jul 2011.
- FLEURY, M. T. L. E. A. **As pessoas na organização.** São Paulo: Gente, 2002.
- GAYLORD, G. L.; RIED, G. E. **Careers in accounting.** [S.l.]: McGraw-Hill Professional, 2006.
- GIL, A. C. **Como elaborar projetos de pesquisa.** São Paulo: Atlas, 1991.
- GIL, A. C. **Métodos e técnicas de pesquisa social.** 5. ed. São Paulo: Atlas, 2007.
- GIL, A. C. **Métodos e técnicas de pesquisa social.** 5. ed. São Paulo: Atlas, 2007.
- GOULART, I. B. **Temas de Psicologia e Administração.** São Paulo: Casa do Psicólogo, 2006.
- HARRELL, C. R.; GHOSH, B. K.; BOWDEN, R. O. **Simulation Using Promodel.** 2nd Edition. ed. Boston: McGraw-Hill, 2003.
- HASHIM, K.; KESHRAF, A. A. **An Approach to Sharing Solutions to Software Project Management Problems.** International Conference on Information Management and Engineering. [S.l.]: [s.n.]. 2009. p. 694 - 697.
- HEINEN, F. **Sistema de Controle Híbrido para Robôs Móveis Autônomos.** UNISINOS. São Leopoldo, p. 130. Dissertação (Mestrado em Computação Aplicada). 2002.
- HERZBERG, F. **Work and Nature Of Man.** Cleveland: The Word Publishing Co., 1966.

- HUGHES, B.; COTTERELL, M. **Software Project Management**. London: McGraw-Hill, 2006.
- HUNT, J. G.; OSBORN, R. N. **Fundamentos de Comportamento Organizacional**. [S.I.]: Bookman, 1999.
- IVANCEVICH, J. M. High and Low Task Stimulation Jobs: A Causal Analysis of Performance-Satisfaction Relationships. **Academy of Management Journal**, v. 22, n. 2, p. 206-222, 1979.
- JENNINGS, N. R.; WOOLDRIDGE, M. Agent-oriented software engineering. In: BRADSHAW, J. **Handbook of Agent Technology**. [S.I.]: AAAI/MIT Press, 2001.
- JENNINGS, N. R.; WOOLDRIDGE, M. J. Applications of Intelligent Agents. In: _____ **Agent Technology: Foundations, Applications and Markets**. New York: Springer-Verlag, 1998. p. 3-28.
- JENNINGS, N.; BUSSMANN, S. Agent-based control systems. **IEEE Computer Systems**, Berlin, Jun. 2003. 61-74.
- JING, H. **Application of Fuzzy Data Mining Algorithm in Performance Evaluation of Human Resource**. International Forum on Computer Science-Technology and Applications. [S.I.]: [s.n.]. 2009. p. 343-346.
- JINGHUA, L.; WENJIAN, L. **An Agent-Based System for Multi-Project Planning and Scheduling**. IEEE International Conference Mechatronics and Automation. [S.I.]: [s.n.]. 1 Jul. 2005. p. 659- 664.
- JO, C.-H.; CHEN, G.; CHOI, J. A framework for BDI Agent-based Software Engineering. **Studia Informatica Universalis**, California, v. 3, n. 3, p. 285-314, 2004.
- JOSLIN, D.; POOLE, W. **Agent-based simulation for software project planning**. Proceedings of the Winter Simulation Conference. Orlando: [s.n.]. 2005. p. 8.
- JUN, P.; QIANGQIANG, Z.; YINGDING, Z. **Application of Multi-Agents System in Intelligent Supervision and Tracing for Agricultural Products Information**. International Conference on E-Product E-Service and E-Entertainment (ICEEE). Henan: [s.n.]. 2010. p. 1-3.
- KERZNER, H. **Project Management**. New Jersey: John Wiley & Sons, 2001.
- KLEIN, G.; JIANG, J.; SOBOL, M. A New View of IS Personal Evaluation. **Communications of the ACM**, v. 44, n. 6, p. 95-101, Jun. 2001.
- LABIDI, S.; LEJOUAD, W. **De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents**. Technical Report 2004, INRIA Institut National de Recherche en Informatique et en Automatique. France. 1993.

- LAPLANTE, P. Remember the human element in IT project management. **IT Professional**, Pennsylvania, v. 5, n. 1, p. 46-50, Jan/Feb 2003.
- LES, K. **Data analysis with Excel: an introduction for physical scientists**. New York: Cambridge University Press, 2002.
- LI, Y. et al. **Modeling e-marketplaces with multi-agents web services**. Proceedings of the 11th International Conference on Parallel and Distributed Systems. [S.l.]: [s.n.]. 2005. p. 175-181.
- LIMA, E. J. C. **ARARA: UM SISTEMA MULTI-AGENTES PARA PROVISÃO DE PERCEPÇÃO EM DESENVOLVIMENTO DE SOFTWARE**. Universidade Federal do Rio de Janeiro. Rio de Janeiro, p. 103. Dissertação (Mestrado em Engenharia de Sistemas e Computação). 2010.
- LIN, H. **Architectural Design of Multi-Agent Systems: Technologies and Techniques**. Downtown: Information Science Reference, 2007.
- LIU, D.; WANG, Q.; XIAO, J. **The role of software process simulation modeling in software risk management: A systematic review**. 3rd International Symposium on Empirical Software Engineering and Measurement. [S.l.]: [s.n.]. 2009. p. 302-311.
- LOPEZ, F. L.; MARQUEZ, A. A. **An Architecture for Autonomous Normative Agents**. Proceedings of Fifth Mexican International Conference in Computer Science. [S.l.]: [s.n.]. 2004. p. 96-103.
- LUSTRI, D.; MIURA, I. K.; TAKAHACHI, S. Gestão do conhecimento desenvolvendo competências. **Revista Inteligência Empresarial**, Rio de Janeiro, n. 25, p. 20-27, Out-Dez 2005.
- MACAL, C. M.; NORTH, M. J. **Tutorial on agent-based modeling and simulation**. Proceedings of the 37th conference on winter simulation. [S.l.]: [s.n.]. 2005.
- MADKIT. Madkit, 2007. Disponível em: <<http://www.madkit.org/>>. Acesso em: 30 jul 2011.
- MASLOW, A. H. **Motivation and personality**. New York: Harper & Brothers, 1954.
- MASON. Multi-Agent Simulator Of Neighborhoods, 2003. Disponível em: <<http://cs.gmu.edu/~eclab/projects/mason/>>. Acesso em: 30 jul 2011.
- MCGREGOR, D. **The Human Side of Enterprise**. New York: McGraw-Hill, 1985.
- MELO, T. Onde está a diferença entre júnior, pleno e sênior?, 2007. Disponível em: <<http://webinsider.uol.com.br/2007/09/18/onde-esta-mesmo-a-diferenca-entre-junior-pleno-e-senior/>>. Acesso em: 3 Jan. 2011.
- MILES, J. A.; KLEIN, H. J. The fairness of assigning group members to tasks. **Group & Organization Management**, v. 23, n. 1, p. 71–96, Mar 1998.

- MING, L.; YANG, N. **Research on the Evaluation System of IT Project**. International Conference on Computational Intelligence and Software Engineering. [S.l.]: [s.n.]. 2009. p. 1-4.
- NBR ISO 1006. **Gestão da Qualidade - Diretrizes para a Qualidade no Gerenciamento de Projetos**. Associação Brasileira de Normas Técnicas. Rio de Janeiro, p. 18. 2000.
- NIENABER, R. C. **A model for enhancing software project management using software agent technology**. University of South Africa. Pretoria, p. 279. (Doctor of Philosophy in Computer Science). 2008.
- ODELL, J.; PARUNAK, H. V. D.; BAUER, B. **Extending UML for Agents**. 17th National Conference on Artificial Intelligence. Austin: [s.n.]. 2000. p. 3-17.
- O'HARE, G. M.; JENNINGS, N. R. **Foundations of Distributed Artificial Intelligence**. New York: Wiley, 1996.
- OLIVEIRA, C. S. **METODOLOGIA PARA UTILIZAÇÃO DE SIMULAÇÃO EM PROJETOS DE MANUFATURA ENXUTA**. Universidade Federal de Minas Gerais. Belo Horizonte, p. 187. Dissertação (Mestrado em Engenharia da Produção). 2008.
- OSÓRIO, F. S. et al. **AMBIENTES VIRTUAIS INTERATIVOS E INTELIGENTES: FUNDAMENTOS, IMPLEMENTAÇÃO E APLICAÇÕES PRÁTICAS**. JAI - SBC. Salvador: [s.n.]. 2004. p. 239-288.
- PADGHAM, L.; WINIKOFF, M. **Prometheus: A Methodology for Developing Intelligent Agents**. First International Joint Conference on Autonomous Agents and Multi-Agent Systems. Bologna: [s.n.]. 2002.
- PEREIRA, I. C. **Proposta de sistematização da simulação para fabricação em lotes**. Universidade Federal de Itajubá. Itajubá, p. 132. Dissertação (Mestrado em Engenharia de Produção). 2000.
- PESSIN, G. et al. Simulação Virtual de Agentes Autônomos para a Identificação e Controle de Incêndios em Reservas Naturais. **IX SVR**, v. 1, p. 236-245, 2007.
- PFLEEGER, S. L. **Engenharia de Software - Teoria e Prática**. 2. ed. São Paulo: Prentice Hall, 2004.
- PHILIPS, J. **IT Project Management: On Track from Start to Finish**. [S.l.]: McGraw Hill Professional, 2010.
- PROJECT MANAGEMENT INSTITUTE. **Um Guia do Conhecimento em Gerenciamento de Projetos - Guia PMBOK**. [S.l.]: Project Management Inc, 2008.
- RAMOS, J. P. **RE-MODELAMENTO DE PROCESSOS E APLICAÇÃO DE TECNOLOGIA DA INFORMAÇÃO PARA GESTÃO DA ALOCAÇÃO DE**

- RECURSOS HUMANOS A UM PORTFÓLIO DE PROJETOS.** Universidade de São Paulo. São Paulo, p. 141. Monografia (Engenharia da Produção). 2009.
- REISS, G. **Project Management Demystified.** London: E&FN Spoon, 1999.
- RIVERA-IBARRA, J. G.; RODRIGUEZ-JACOBO, J.; SERRANO-VARGAS, M. A. **Competency Framework for Software Engineers.** Conference on Software Engineering Education and Training (CSEE&T). [S.l.]: [s.n.]. 2010. p. 33-40.
- ROBINSON, S. **Simulation: The Practice of Model Development and Use.** England: John Wiley & Sons Ltd, 2004.
- SCHNAIDER, L. R. C. **PLANEJAMENTO DA ALOCAÇÃO DE RECURSOS HUMANOS EM AMBIENTES DE DESENVOLVIMENTO DE SOFTWARE ORIENTADOS À ORGANIZAÇÃO.** Universidade Federal do Rio de Janeiro. Rio de Janeiro, p. 133. Dissertação (Mestrado em Engenharia de Sistemas e Computação). 2003.
- SCHWALBE, K. **Introduction to Project Management.** 2. ed. [S.l.]: [s.n.], 2008.
- SEILA, A. F. **Introduction to Simulation.** Proceeding of Winter Simulation. Arlington, USA: [s.n.]. 1995. p. 7-15.
- SELLTIZ, C. et al. **Métodos de pesquisa nas relações sociais.** 2. ed. São Paulo: Editora Herder, 1967.
- SESAM. Shell for Simulated Agent Systems., 2009. Disponível em: <<http://www.simsesam.de/>>. Acesso em: 30 jul 2011.
- SIEBERS, P. O.; AICKELIN, U. Introduction to multi-agent simulation, 2007. Disponível em: <<http://arxiv.org/ftp/arxiv/papers/0803/0803.3905.pdf>>. Acesso em: 7 Jan. 2011.
- SILVA, F. et al. **Um Modelo de Simulação de Processos de Software baseado em Agentes Cooperativos.** XIII Simpósio Brasileiro de Engenharia de Software. Florianópolis: [s.n.]. 1999. p. 163-178.
- SILVA, I. G. **Projeto e Implementação de Sistemas Multi-Agentes: O Caso Tropos.** Universidade Federal de Pernambuco. Recife, p. 108. Dissertação (Mestrado em Ciência da Computação). 2005.
- SILVA, L. C. AGAIS. **Simulação de Processos,** 2005. Disponível em: <<http://www.agais.com/simula.htm>>. Acesso em: 12 Feb. 2011.
- SILVERSTEIN, B. **Motivação: Desperte o que há de melhor em sua equipe.** Rio de Janeiro: Senac, 2009.
- SIMCOG. Simulation of Cognitive Agents, 2002. Disponível em: <<http://www.lti.pcs.usp.br/SimCog/>>. Acesso em: 30 ju 2011.

- SIMULA. The Simula Programming Language., 1996. Disponível em: <<http://groups.engin.umd.umich.edu/CIS/course.des/cis400/simula/simula.html>>. Acesso em: 30 jul 2011.
- SIROTA, D.; MISCHKIND, L. A.; MELTZER, M. I. **The Enthusiastic Employee: How Companies Profit by Giving Workers What They Want.** [S.I.]: Wharton School Publishing, 2005.
- SLOMAN, A. The SimAgent TOOLKIT, 2005. Disponível em: <<http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html>>. Acesso em: 30 jul 2011.
- SOFTWARE ENGINEERING INSTITUTE. CMMI, 2011. Disponível em: <<http://www.sei.cmu.edu/cmmi/>>. Acesso em: 3 Jan. 2011.
- SOMMERVILLE, I. **Engenharia de Software.** São Paulo: Pearson Addison-Wesley, 2007.
- SOMMERVILLE, I.; RODDEN, T. Human, social and organizational influences on the software process. In: FUGGETTA, A.; WOLF, A. **Trends in Software.** [S.I.]: John Wiley & Sons, v. 4, 1996. p. 1-21.
- SPIDERLAND. BREVE, 2002. Disponível em: <<http://www.spiderland.org>>. Acesso em: 17 jul. 2011.
- STANDISH GROUP. The CHAOS Report, Boston, 2009. Disponível em: <http://www1.standishgroup.com/newsroom/chaos_2009.php>. Acesso em: 17 Jun. 2010.
- STARLOGO. STARLOGO, 2006. Disponível em: <<http://education.mit.edu/starlogo/>>. Acesso em: 30 jul 2011.
- STREIT, R. E. **Um Modelo Baseado em Agentes para a Análise da Governança Regulamentar do Sistema Financeiro.** Universidade Federal do Rio Grande do Sul. Porto Alegre, p. 286. Tese (Doutorado em Administração). 2006.
- SWARM. SWARM, 2003. Disponível em: <<http://www.swarm.org/>>. Acesso em: 30 jul 2011.
- TWEEDALE, J. et al. Innovations in multi-agent systems. **Journal of Network and Computer Applications**, v. 30, n. 3, p. 1089-1115, Aug 2007.
- VARGAS, R. Gestão de Projetos, 2011. Disponível em: <<http://www.ricardo-vargas.com/slides/22/>>. Acesso em: 3 Jan. 2011.
- WANG, Q. et al. A Process-Agent Construction Method for Software Process Modeling in SoftPM. **SPW/ProSim**, Shanghai China, v. 3966, p. 204-213, 2006.

- WANG, Q.; LAI, X. **Requirements Mangement for the Increment Development Model**. Proceedings of the Second Asia-Pacific Conference on Quality Software. [S.l.]: [s.n.]. 2001.
- WANG, S.-Q.; GONG, L.-H.; YAN, S.-L. **The Allocation Optimization of Project Human Resource Based on Particle Swarm Optimization Algorithm**. IITA International Conference on Services Science, Management and Engineering. Zhangjiajie : [s.n.]. 2009. p. 168-172.
- WANG, X.; JIANG, Y. **The Study of Enterprise Human Resource Evaluation Research Based on Neural Network**. Third International Symposium on Information Processing (ISIP). [S.l.]: [s.n.]. 2010. p. 19-21.
- WENG, W. et al. **An Approach for Allocation Optimization of Multi-Project Human Resource Based on DEA**. International Conference on Management and Service Science (MASS). [S.l.]: [s.n.]. 2010. p. 1-4.
- WENG, Y.; CHEN, J.; ZHAO, S. Software Project Management Evaluation Based on Evidence Theory. **Fourth International Conference on Computer Sciences and Convergence Information Technology**, Seoul , 24-26 Nov. 2009. 632.
- WICKENBERG, T.; DAVIDSSON, P. **On Multi Agent Based Simulation of Software Development Processes**. 3rd international conference on Multi-agent-based simulation II. [S.l.]: [s.n.]. 2002. p. 104-113.
- WILENSKY, U. Netlogo, 1999. Disponivel em: <<http://ccl.northwestern.edu/netlogo/>>. Acesso em: 30 jul 2011.
- WOHLIN, C. Are Individual Differences in Software Development Performance Possible to Capture Using a Quantitative Survey? **Empirical Software Engineering**, v. 9, n. 3, p. 211-228, 2004.
- WOOD JR., T.; PICARELLI FILHO, V. **Remuneração e carreira por habilidades e competências**: preparando a organização para a era das empresas de conhecimento intensivo. 3. ed. São Paulo: Atlas, 2004.
- WOOD, M. F.; DELOACH, S. A. **An Overview of the Multiagent Systems Engineering Methodology**. The First International Workshop on Agent-Oriented Software Engineering. Limerick: [s.n.]. 2000. p. 207-222.
- WOOLDRIDGE, M. **An Introduction to Multiagents Systems**. Chichester: John Wiley & Sons LTD, 2002.
- WU, M. et al. Balanced reactive-deliberative architecture for multi-agent system for simulation league of RoboCup. **International Journal of Control, Automation and Systems**, v. 7, n. 6, p. 945-955, Dec 2009.
- XIAO, J. et al. **A Constraint-Driven Human Resource Scheduling Method in Software Development and Maintenance Process**. IEEE International Conference on Software Maintenance, ICSM. Beijing: [s.n.]. 2008. p. 17.

XIE, F.; TANG, Q. **Human Resource Development by Fuzzy Neural Networks**. 4th International Conference on Wireless Communications, Networking and Mobile Computing. [S.l.]: [s.n.]. 2008. p. 1-4.

ZHOU, L. **A Project Human Resource Allocation Method Based on Software Architecture and Social Network**. International Conference on Wireless Communications, Networking and Mobile Computing. [S.l.]: [s.n.]. 2008. p. 1-6.

ANEXOS

ANEXO A – Dados do Projeto

Atividades - Projeto 1									
ID	Tipo	Atividades Predecessoras	Nome	Habilidades	Nível		Data de Início	Data de Entrega	Orçamento
1	Análise de Requisitos	0	P1T1	Análise Orientada a Objetos	3	Senior	2007-05-28	2007-06-20	25000
2	Projeto de Software	1	P1T2	Projeto Orientado a Objetos	4	Master	2007-06-21	2007-07-13	12000
3	Projeto de Software	1	P1T3	Projeto Orientado a Objetos	3	Senior	2007-06-21	2007-07-06	10000
4	Programação	2	P1T4	LingProg - Linguagem Java	4	Master	2007-07-16	2007-08-17	24000
				Ferramentas - Eclipse	2	Pleno			
5	Programação	3	P1T5	LingProg - Linguagem Java	3	Senior	2007-07-09	2007-08-17	15000
				Ferramentas - Eclipse	2	Pleno			
6	Testes	1	P1T6	Testes - Escrever casos de teste	4	Master	2007-06-21	2007-08-17	25000
7	Testes	4, 5, 6	P1T7	Testes - Executar casos de teste	4	Master	2007-08-20	2007-10-12	32000

Fonte: Adaptada de Xiao *et al.* (2008).

ANEXO B – Dados dos Recursos Humanos

Recursos Humanos										
ID	Tipo	Experiencia	Nome	Habilidades	Nível		Data de Início	Data Final	N de Horas/dia	\$Homem/hora
1	Análise de Requisitos	0,0625	jh	Análise Orientada a Objetos	4	Master	2007-05-08	2007-07-30	4	60
				Análise Estruturada	3	Senior				
2	Análise de Requisitos	0,05	hls	Análise Orientada a Objetos	3	Senior	2007-05-28	2007-09-28	8	45
3	Análise de Requisitos	0,06	cy	Análise Orientada a Objetos	4	Master	2007-05-08	2007-08-31	6	60
	Projeto de Software	0,06		Projeto orientado a Objeto	4	Master				
	Programação	0,03		LingProg - Programação Java	4	Master	2007-09-03	2007-10-31	8	
				Ferramentas - Eclipse	4	Master				
4	Análise de Requisitos	0,05	wht	Análise Orientada a Objetos	4	Master	2007-05-08	2007-10-31	6	50
	Projeto de Software	0,06		Projeto orientado a Objeto	4	Master				
	Programação	0,03		LingProg - Programação Java	4	Master				
				Ferramentas - Eclipse	4	Master				
5	Projeto de Software	0,06	wzg	Projeto orientado a Objeto	3	Senior	2007-05-08	2007-09-14	8	50
	Programação	0,035		LingProg - Programação Java	3	Senior				
				LingProg - Programação C#	3	Senior				
				Ferramentas - Eclipse	4	Master				
6	Projeto de Software	0,05	yjj	Projeto orientado a Objeto	3	Senior	2007-05-08	2007-09-14	8	45
	Programação	0,025		LingProg - Programação Java	4	Master				
				Ferramentas - Eclipse	4	Master				
7	Projeto de Software	0,04	dj	Projeto orientado a Objeto	2	Pleno	2007-05-08	2007-08-31	8	20
	Programação	0,025		LingProg - Programação Java	4	Master				
				Ferramentas - Eclipse	3	Senior				


8	Projeto de Software	0,04	fc	Projeto orientado a Objeto	2	Pleno	2007-05-08	2007-08-31	8	20
	Programação	0,02		LingProg - Programação Java	4	Master				
				Ferramentas - Eclipse	3	Senior				
9	Projeto de Software	0,04	zrm	Projeto orientado a Objeto	2	Pleno	2007-05-08	2007-08-31	8	20
	Programação	0,025		LingProg - Programação Java	3	Senior				
				Ferramentas - Eclipse	3	Senior				
10	Testes	0,05	hcy	Testes - Escrever casos de teste	4	Master	2007-05-08	2007-10-31	8	40
		0,04		Testes - Executar casos de teste	4	Master				
11	Testes	0,035	lt	Testes - Escrever casos de teste	4	Master	2007-05-08	2007-11-30	8	45
		0,03		Testes - Executar casos de teste	4	Master				
12	Testes	0,035	yf	Testes - Escrever casos de teste	3	Senior	2007-05-08	2007-10-31	8	40
		0,03		Testes - Executar casos de teste	3	Senior				
13	Testes	0,035	cp	Testes - Escrever casos de teste	3	Senior	2007-05-08	2007-10-31	8	40
		0,03		Testes - Executar casos de teste	3	Senior				
14	Testes	0,03	lq	Testes - Executar casos de teste	3	Senior	2007-05-08	2007-10-31	8	20

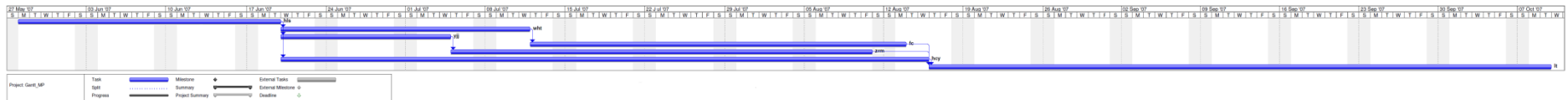
Fonte: Adaptada de Xiao et al. (2008).

APÊNDICE A – Dados de Aprendizagem



ID	Recurso	Habilidade	Nivel		Aprendizado
5	wzg	Projeto orientado a Objeto	3	Senior	S
8	fc	Projeto orientado a Objeto	2	Pleno	S
9	zrm	Projeto orientado a Objeto	2	Pleno	S
9	zrm	LingProg - Programação Java	3	Pleno	S
12	yf	Testes - Executar casos de teste	3	Pleno	S
13	cp	Testes - Executar casos de teste	3	Pleno	S
14	lq	Testes - Executar casos de teste	3	Pleno	S

APÊNDICE B – Planejamento Melhor Prazo



ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		P1T1	17 days	Mon 28-05-07	Tue 19-06-07		hls
2		P1T2	16 days	Wed 20-06-07	Wed 11-07-07	1	wht
3		P1T3	11 days	Wed 20-06-07	Wed 04-07-07	1	yjj
4		P1T4	23 days	Thu 12-07-07	Mon 13-08-07	2	fc
5		P1T5	27 days	Thu 05-07-07	Fri 10-08-07	3	zrm
6		P1T6	41 days	Wed 20-06-07	Wed 15-08-07	1	hcy
7		P1T7	39 days	Thu 16-08-07	Tue 09-10-07	4,5,6	lt





APÊNDICE C – Planejamento Melhor Custo

ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		P1T1	14 days	Mon 28-05-07	Thu 14-06-07		wht
2		P1T2	23 days	Fri 15-06-07	Tue 17-07-07	1	zrm
3		P1T3	14 days	Fri 15-06-07	Wed 04-07-07	1	dj
4		P1T4	23 days	Wed 18-07-07	Fri 17-08-07	2	fc
5		P1T5	24 days	Thu 05-07-07	Tue 07-08-07	3	dj
6		P1T6	41 days	Fri 15-06-07	Fri 10-08-07	1	hcy
7		P1T7	39 days	Mon 20-08-07	Thu 11-10-07	4,5,6	hcy

APÊNDICE D – Planejamento Aprendizado 30%

ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		P1T1	17 days	Mon 28-05-07	Tue 19-06-07		hls
2		P1T2	20 days	Wed 20-06-07	Tue 17-07-07	1	wzg
3		P1T3	14 days	Wed 20-06-07	Mon 09-07-07	1	zrm
4		P1T4	23 days	Wed 18-07-07	Fri 17-08-07	2	dj
5		P1T5	27 days	Tue 10-07-07	Wed 15-08-07	3	zrm
6		P1T6	41 days	Wed 20-06-07	Wed 15-08-07	1	hcy
7		P1T7	47 days	Mon 20-08-07	Tue 23-10-07	4,5,6	yf

APÊNDICE E – Planejamento Aprendizado 100%

ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		P1T1	14 days	Mon 28-05-07	Thu 14-06-07		jh
2		P1T2	20 days	Fri 15-06-07	Thu 12-07-07	1	wzg
3		P1T3	14 days	Fri 15-06-07	Wed 04-07-07	1	zrm
4		P1T4	21 days	Fri 13-07-07	Fri 10-08-07	2	cy
5		P1T5	27 days	Thu 05-07-07	Fri 10-08-07	3	zrm
6		P1T6	50 days	Fri 15-06-07	Thu 23-08-07	1	yf
7		P1T7	47 days	Fri 24-08-07	Mon 29-10-07	4,5,6	yf