

# WEBSMIT – Sistema Multiagente para Identificação e Atribuição Automática de Tarefas para Grupos de Trabalho Geograficamente Dispersos

César Arboite Pereira<sup>1</sup> José Lucas Garcia Filho<sup>1</sup> Márcia Cristina Moraes<sup>1</sup> & Silvia Moraes<sup>1</sup>

<sup>1</sup>Faculdade de Informática  
Pontifícia Universidade Católica do Rio Grande do Sul  
Av. Ipiranga 6681, Prédio 32  
Telefone: +55 (51) 33203558  
Porto Alegre - RS - BRAZIL  
{cesarap-lucasjava}@gmail.com {marcia.moraes-silvia.moraes}@pucrs.br

## Abstract

*Software projects that involve multiple work teams have a greater complexity than other types of projects, because there are many factors that project managers need to consider. In geographically distributed projects, each team may be responsible for implementing a part of the project, and considering that they are geographically distributed it is possible to have some interaction difficulties.*

*A system that uses multi-agent systems for automatic allocation of work teams is an interesting alternative. Such a tool helps project manager in the allocation of teams without the need of the manager to have a detailed knowledge about the skills of the available resources for selecting a team.*

*This paper presents a prototype tool, called WEBSMIT (Multi-Agent System for Automatic Identification and Assignment of Tasks for Teams*

*Geographically Distributed), which identify the best team available to perform the activity required for a project.*

**Keywords:** Allocation of tasks, Geographically distributed teams, Multi-agent systems, Analogy by similarity

## 1. INTRODUÇÃO

Atualmente, o projeto de um novo produto tem se mostrado um processo intensivamente colaborativo. Isto ocorre devido à complexidade dos projetos que envolvem várias equipes de trabalho, cada uma responsável pela execução de uma parte do projeto. Estas equipes, em muitos casos, estão localizadas em pontos geograficamente distantes o que dificulta a interação entre elas pela ausência de um contato mais direto e pessoal.

Neste caso, algumas situações particulares podem

surgir e influenciar diretamente o desenvolvimento do projeto [1]. Uma possível situação é quando ocorre uma falha (*bug*) no produto e essa falha tem nuances de diferentes equipes; o problema, então, passa a ser a identificação da equipe mais adequada para solucionar tal falha. A mesma situação pode acontecer quando é necessário realizar uma extensão das funcionalidades do produto, antes ou após a sua entrega. Essa extensão pode exigir o envolvimento de diferentes equipes. No entanto, o inter-relacionamento das atividades necessárias ao seu desenvolvimento pode dificultar a divisão e distribuição dessas atividades entre as equipes.

Muitas vezes, com gerente de projetos e equipes de desenvolvimento trabalhando em locais geograficamente distantes, se torna difícil para o gerente de projetos conhecer profundamente as equipes de trabalho e tomar a decisão de alocar a equipe mais adequada para o desenvolvimento de alguma tarefa.

A fim de auxiliar o gerente de projetos no processo de alocação de equipes, geograficamente dispersas, que possuam as competências necessárias para realizarem atividades de um projeto, estamos propondo o sistema WEBSMIT (Sistema Multiagente para Identificação e Atribuição Automática de Tarefas para Grupos de Trabalho Geograficamente Dispersos). Este sistema faz uso do paradigma de sistemas multiagentes para identificar a disponibilidade e as competências de uma equipe e alocá-la para a realização de uma atividade de um projeto.

O artigo está organizado em quatro seções. A seção dois descreve o referencial teórico utilizado neste trabalho. A seção três apresenta a modelagem do sistema proposto e o seu funcionamento. A seção quatro descreve as considerações finais e trabalhos futuros.

## 2. REFERENCIAL TEÓRICO

Para o desenvolvimento deste trabalho foi necessário o estudo de conceitos relacionados às áreas de engenharia de software e de inteligência artificial. Os conceitos estudados serão apresentados nas próximas subseções.

### 2.1 PROJETO DE SOFTWARE

De acordo com o PMBOK [2], projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. A principal diferença entre projetos e trabalho operacional, embora possuam características semelhantes, é o fato de que as operações são contínuas e repetitivas, enquanto que os projetos são temporários e exclusivos.

### 2.2 PROJETOS GEOGRAFICAMENTE DISTRIBUÍDOS

Projetos geograficamente distribuídos são aqueles que possuem uma ou mais equipes alocadas em locais

diferentes, tais como cidades, estados e países. Muitas organizações optam por esta abordagem por se tratar, muitas vezes, de uma oportunidade de aumento nos lucros, uma vez que se pode obter mão-de-obra qualificada por menores custos, diminuição nos índices de retrabalhos, além de utilizar, muitas vezes, a diferença no fuso horário a favor do projeto. Em contrapartida, esta abordagem tem como consequência uma série de riscos que não se teria em um ambiente de projeto convencional. Podemos citar como exemplos a diferença sócio-cultural entre as equipes, a comunicação e o relacionamento entre as equipes.

Nesse contexto desafiador, cabe ao gerente de projetos fazer com que suas equipes trabalhem em harmonia, visando o cumprimento dos prazos estabelecidos, dentro do custo e do escopo previsto, e com a qualidade esperada pelo cliente. Fica visível que neste cenário torna-se praticamente impossível gerenciar sem o apoio de ferramentas que dêem subsídios para tomadas de decisões ao gerente do projeto [3].

### 2.3 GERENCIAMENTO DE RECURSOS HUMANOS

Conforme o PMBOK [2], o gerenciamento dos recursos humanos do projeto inclui os processos que organizam e gerenciam a equipe do projeto. A equipe do projeto é composta por pessoas com funções e responsabilidades atribuídas para o término do projeto. Os recursos humanos do projeto são todos os membros do projeto que serão designados a exercerem determinadas funções e responsabilidades no projeto, de acordo com suas competências. Função é o rótulo que descreve a parte de um projeto pela qual uma pessoa é responsável [2]. Em projetos de software as funções também são conhecidas como papéis. São exemplos de funções em equipes de projetos de software: analista de sistemas, analista de negócios, gerente de projetos, desenvolvedor, analista de testes, etc. Responsabilidade é o trabalho que um membro da equipe do projeto deve realizar para terminar as atividades do projeto. Como exemplos de responsabilidades em projetos de software podemos citar: codificação dos casos de uso, realização de testes funcionais dos casos de uso, especificação de casos de uso, entre outras. Competências são as habilidades e as capacidades necessárias para realizar as atividades do projeto [2].

### 2.4 AGENTES E SISTEMAS MULTIAGENTES

Em Inteligência Artificial, é difícil encontrar uma única definição para o termo agentes, porém, existe um consenso de que autonomia é a característica central de todos os tipos de agentes. Parte da dificuldade de se encontrar uma definição geral se deve a existência de diferentes objetivos para diferentes situações em que se utilizam agentes. Segundo Weiss [4], uma definição abrangente sobre o termo agente diz que ele pode ser

considerado como uma entidade situada em um ambiente capaz de tomar decisões autônomas e ter um comportamento específico a fim de realizar os objetivos para aquilo que foi projetado.

Um agente para conseguir realizar seu objetivo, deve possuir algumas propriedades específicas como: ser autônomo, para conseguir operar sem a direta intervenção humana; social, para conseguir cooperar com humanos ou com outros agentes para cumprir tarefas; reativo, para responder às percepções de mudança de ambiente; e pró-ativo para não simplesmente responder as mudanças de ambiente, mas para tomar iniciativas rumo ao objetivo [5].

Quando um conjunto de agentes interage em um ambiente comum temos o que se denomina de sistema multiagente [6]. De acordo com Weiss [4] as características de um sistema multiagente são: a existência de uma infra-estrutura de comunicação entre os agentes; os sistemas são normalmente abertos e compostos por agentes autônomos e distribuídos que podem ser cooperativos ou competitivos.

### 2.5 ANALOGIA POR SIMILARIDADE

Uma abordagem que fornece respostas rápidas e usa experiência passada (recursos) para produzir novas soluções para problemas inesperados é a analogia por similaridade [7]. Um modelo de analogia simplificado em um banco de dados é o seguinte: existe um alvo T descrito por  $m$  pares de atributo-valor, para o qual se quer encontrar o valor de outro atributo Q. Existem várias fontes  $S_1, \dots, S_n$  (análogas) que têm valores para o atributo Q desejável bem como para os  $m$  atributos conhecidos para o alvo. A similaridade  $s$  é definida como o número de valores de atributos que combinam para um determinado alvo e fonte e a diferença é definida por  $d = m - s$ . Assumindo que existem  $r$  atributos relevantes para descobrir o valor de Q,  $p(d, r)$  é definida como sendo a probabilidade que a fonte S, diferindo do seu alvo em  $d$  atributos, se combina com os  $r$  atributos relevantes. A suposição de que não existe informação relevante significa que todos os atributos são igualmente importantes para serem relevantes. Pode-se calcular  $p(d, r)$  usando um argumento combinatório simples (Russell 1988). Seja  $N_m$  o número de escolhas de quais atributos são relevantes tais que S combina com T em tais atributos. Seja  $N$  o número total de escolhas de quais atributos são relevantes, temos a seguinte equação (1):

$$p(d, r) = N_m / N = \frac{\binom{m-d}{r}}{\binom{m}{r}} \quad (r \geq 1) \quad (1)$$

## 3. SISTEMA WEBSMIT

O sistema WEBSMIT é um sistema Web que tem

como objetivo auxiliar o gerente de projetos no processo de encontrar os melhores recursos disponíveis para a realização de atividades em um projeto de software geograficamente distribuído. O sistema é composto por três módulos (Figura 1): segurança (responsável pela validação de usuários); cadastros (responsável por popular a base de dados do sistema com informações relevantes para o processo de alocação de recursos) e agentes (responsável pelo processo de seleção e alocação de equipes para atividades do projeto). Na s próximas seções serão detalhadas as informações armazenadas na base de dados, o módulo de agentes e o funcionamento do sistema.

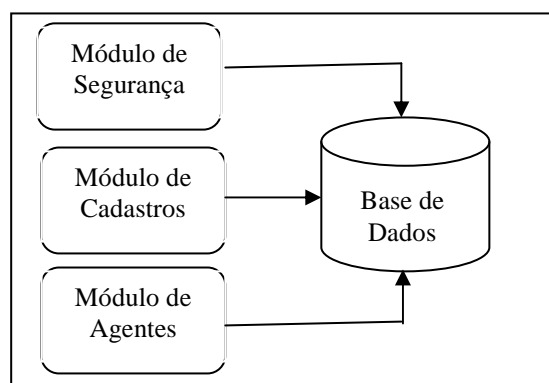


Figura 1: Módulos do sistema

### 3.1 INFORMAÇÕES ARMazenADAS NA BASE DE DADOS

Como apresentado na Figura 1, para o desenvolvimento do sistema WEBSMIT será necessário possuir uma base de dados com informações de projetos, atividades, tarefas, equipes e recursos. O sistema usa como base para realizar a alocação das equipes as competências de cada recurso e para isso, portanto, também será necessário armazenar informações sobre conhecimentos específicos de cada integrante de uma equipe.

De acordo com estas necessidades, a seguir serão apresentadas as informações que armazenaremos na base de dados do sistema.

- Projeto

Não existe atividade se esta não está atrelada a um projeto, logo, o cadastro de projetos é o primeiro passo do sistema. Um projeto deverá possuir um nome e uma descrição para ser identificado. O campo descrição será utilizado somente para identificação do projeto em buscas posteriores ao cadastramento. Além disso, os projetos serão constituídos por uma série de atividades.

- Atividade

Como citado anteriormente, uma atividade fará parte de um projeto, e para estas atividades é que o sistema irá

realizar a alocação da equipe mais adequada. Cada atividade possuirá uma descrição e um atributo indicando a qual projeto está relacionada. Como a alocação da equipe será feita por atividade, teremos também que armazenar a equipe que estará realizando a atividade após ser feita a alocação automática.

- Tarefa

Definimos que as tarefas em um projeto, pertencem a uma atividade e são realizadas por um único recurso, que possui o papel requerido para a tarefa e também as competências necessárias para realizá-la. A alocação automática irá utilizar estas informações de competências para buscar o melhor recurso para a tarefa.

- Recurso

Um recurso em um projeto representa uma pessoa. Cada recurso possui uma série de competências que podem ou não ser relacionadas ao seu papel, como, por exemplo, um recurso que possui um papel de programador pode possuir competência em UML, que também é uma competência do papel de analista de sistemas.

Após a alocação automática, um recurso será relacionado a uma tarefa, indicando que a tarefa está sendo realizada. Esta informação será futuramente armazenada em uma base de histórico de atividades da equipe, para que o sistema, quando for fazer uma nova alocação verifique se o recurso e a equipe já realizaram uma tarefa relacionada.

- Equipe

Uma equipe será formada por recursos de diferentes competências. Com base nas competências dos recursos pertencentes à equipe o sistema realizará a alocação da equipe em uma atividade.

- Competência

As competências são utilizadas para alocar os recursos às tarefas e conseqüentemente as equipes às atividades. Cada recurso poderá possuir  $n$  competências.

Uma competência não estará diretamente ligada a um papel, visto que um programador pode, por exemplo, possuir o conhecimento de PMBOK, apesar de não fazer uso dela enquanto realiza o papel de programador. Esta visão facilita a alteração do papel de um recurso, que pode a qualquer momento receber outro papel, ou seja, o recurso passa a exercer um novo papel, mas continuará com as competências que utilizava para desempenhar seu papel anterior.

Como exemplos de competências podemos citar: Java, C#, Sql ANSI, Oracle, SqlServer, Php, Eclipse,

Visual Studio, Html, Javascript, UML, Webservice, XML, Ajax, CSS, PL/SQL, PMBOK, Scrum, Liderança, Comunicação, Negociação, Inglês e Espanhol.

- Papel

Os papéis são as funções exercidas pelos membros da equipe (recursos) durante um projeto. Um recurso poderá exercer apenas um papel.

Alguns dos papéis que podem ser executados pelos recursos de uma equipe, sendo que uma equipe poderá tanto possuir mais de um recurso para cada papel, como nenhum recurso para outro papel, são: programador, testador, analista de sistemas, gerente de projetos, DBA (*Data Base Administrador*) e arquiteto de software.

### 3.2 MÓDULO DE AGENTES

Para o desenvolvimento do módulo de agentes foi utilizado um sistema multiagente composto pelos seguintes agentes: agente de grupo, agente seletor de grupos e agente gerenciador de tarefas, como mostra a Figura 2. Estes agentes fazem acesso a uma base de dados geral de atividades, que contém tanto as atividades realizadas pelas equipes quanto às atividades que estão em andamento.

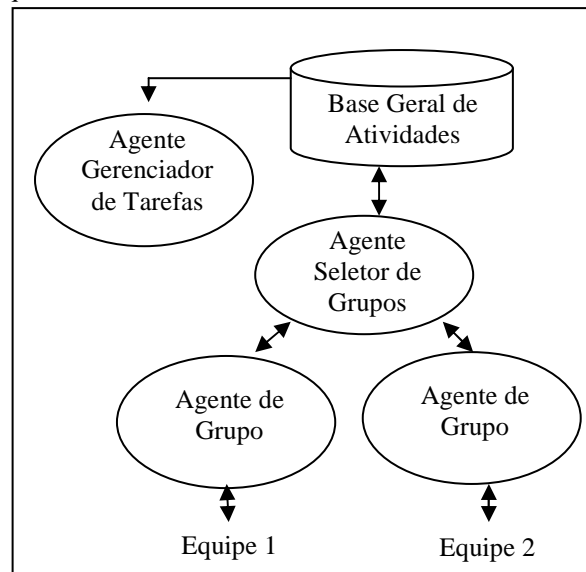


Figura 2: Arquitetura do módulo de agentes

Agente de grupo é aquele que representa virtualmente uma equipe engajada no desenvolvimento do projeto. O agente tem o conhecimento e controle sobre as atividades que estão sob sua responsabilidade e da sua equipe. Com base nestas informações o agente informa, quando solicitado, um índice probabilístico de aptidão da equipe que define a capacidade da equipe executar uma determinada atividade. Para realizar o cálculo deste índice, o agente verifica quais tarefas estão cadastradas na atividade a ser realizada. As tarefas

necessitam sempre de um recurso com as competências adequadas para realizá-las. O papel do agente de grupo é verificar se todas as tarefas da atividade têm condições de serem cumpridas pelos recursos de sua equipe. Esta verificação é realizada utilizando-se o algoritmo de analogia por similaridade apresentado na seção 2.5.

Agente seletor de grupos é responsável pela escolha do grupo (que representa uma equipe) mais adequado para resolver uma atividade. Este agente recebe uma solicitação de um agente de grupo quanto à escolha de uma equipe para resolver uma determinada tarefa, este agente realiza uma busca na base geral das atividades e verifica se a tarefa já foi resolvida ou se sua resolução está em andamento. Caso ele não a encontre, ele dispara um *broadcast* para todos os agentes de grupo solicitando que eles informem seus índices de aptidão e disponibilidade para executar tal tarefa. Ao receber as respostas dos agentes por grupo quanto à execução de uma tarefa, este agente decide, através de uma lista ordenada, qual agente de grupo resolverá a tarefa e informa ao agente a sua decisão.

Agente gerenciador de tarefas é o responsável pelo controle e atualização das atividades realizadas por todos os agentes de grupo na base geral das atividades

do projeto. Ele garante a consistência da base gerh de atividades, evitando, por exemplo, que a mesma atividade esteja em execução por grupos diferentes. Ao receber uma requisição do agente de grupo para registrar uma tarefa em execução na base geral das atividades, o agente verifica se a atividade a ser registrada já está sendo executada por outro grupo. Neste caso, este agente informa ao agente de grupo solicitante que a tarefa já está em execução e não realiza o registro. No outro caso, o agente atualiza a base de atividades e responde ao agente de grupo solicitante uma confirmação da sua solicitação. Ao receber uma requisição do agente de grupo para registrar o término de uma tarefa na base geral das atividades, ele atualiza o registro correspondente à atividade e responde confirmando a solicitação.

Para realizar a comunicação entre os agentes foi desenvolvido um protocolo baseado em troca de mensagens. Este protocolo pode ser visualizado na Figura 3.

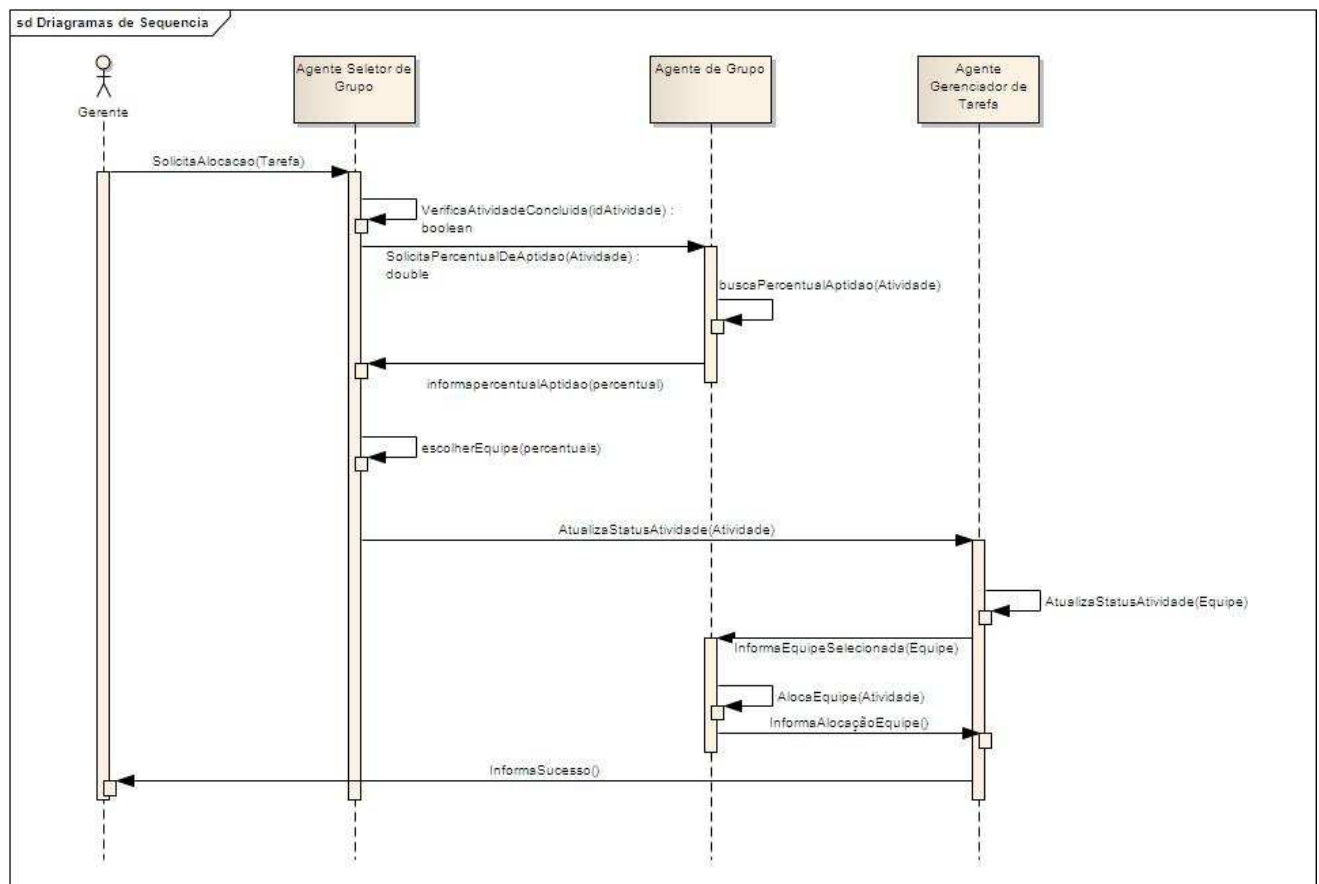


Figura 3: Protocolo de comunicação entre os agentes

Como apresentado na Figura 3, o gerente de projetos solicita pelo sistema WEBSMIT a alocação de uma equipe para realizar uma determinada tarefa. O agente que é ativado neste momento é o agente seletor de grupos. Este agente realiza uma busca na base geral das atividades e verifica se a tarefa já foi resolvida ou se sua resolução está em andamento. Caso a tarefa não esteja na base de dados, o agente seletor de grupo dispara um *broadcast* para todos os agentes de grupo solicitando os seus índices de aptidão para executar a tarefa. Os agentes de grupo realizam o cálculo do seu índice de aptidão utilizando o algoritmo de analogia por similaridade (este processo será detalhado no próximo parágrafo) e informa este índice para o agente seletor de grupo. Após ter recebido todos os índices de aptidão de cada agente de grupo, o agente seletor de grupo cria uma lista ordenada de grupos que podem realizar a tarefa e faz a seleção do melhor grupo. Tendo selecionado um grupo para realizar a tarefa, o agente seletor de grupos envia uma mensagem para o agente gerenciador de tarefas que atualiza o estado da atividade na base de dados geral de atividades. O agente gerenciador de tarefas envia uma mensagem para o agente de grupo da equipe selecionada. O agente de grupo aloca a equipe e informa ao agente gerenciador de tarefas que a sua equipe foi alocada para realizar a tarefa. O agente gerenciador de tarefas informa ao gerente de projeto o resultado da sua solicitação.

A alocação automática de recursos feita pelos agentes desenvolvidos levará dois pontos em consideração: as competências necessárias para a realização de cada tarefa pertencente à atividade; e as competências dos recursos pertencentes à equipe e aptos a realizar as tarefas da atividade.

Quando uma atividade é cadastrada, o gerente de projetos também cadastra tarefas para a atividade. Ao cadastrar uma tarefa, o gerente de projetos estabelece um papel que deverá executá-la e as competências necessárias para a execução sua execução. Cada tarefa poderá ser executada por somente um recurso, e uma atividade deverá ser executada somente por uma equipe, ou seja, todas as tarefas de uma atividade serão executadas por recursos de uma mesma equipe. Para verificar o percentual de aptidão de cada recurso e por consequência da equipe, o agente de grupo verifica se algum recurso da equipe executa o papel necessário para a tarefa e se possui as competências necessárias. Além disso, a disponibilidade dos recursos também é analisada, ou seja, se um recurso estiver alocado para outra tarefa no mesmo período tarefa solicitada, este não poderá ser superalocado

Para realizar o cálculo do percentual de aptidão da

para a atividade, é utilizado o algoritmo de analogia por similaridade, como mencionado anteriormente.

Com as atividades, tarefas e competências devidamente cadastradas temos então o seguinte cenário a ser levado em consideração para o cálculo do percentual de aptidão:

$m \rightarrow$  número de competências necessárias para execução da tarefa

$s \rightarrow$  número de combinações da lista de competências de um recurso com a lista de competências da tarefa

$d = m - s \rightarrow$  diferença entre a lista de competências desejadas para a execução da tarefa e as competências possuídas pelo recurso

$r = m/2 \rightarrow$  número de atributos relevantes, ou seja, definimos que o recurso deve possuir no mínimo 50% das competências necessárias para ser considerado apto.

Após a aplicação das variáveis na equação (1) apresentada na seção 2.5, será gerado um número entre 0 e 1 que irá determinar a aptidão de um recurso poder realizar a tarefa.

### 3.3 FUNCIONAMENTO DO SISTEMA

Apresentaremos a seguir os passos necessários para a execução do fluxo principal de utilização do sistema, desde o acesso ao sistema, ao cadastramento de projeto, atividades, recursos, papéis, competências e equipes.

Após o gerente de projetos ter realizado o *login* no sistema, ele terá acesso a “Tela Inicial”, como apresentado na Figura 4.



Figura 4: Tela inicial do sistema

A seguir demonstraremos o fluxo de cadastro de um projeto. Apesar de começarmos pelo cadastro de projeto neste exemplo, esclarecemos que poderíamos ter iniciado por outros cadastros, como, por exemplo,

o cadastro de competências.

Ao clicar no link “Projetos” do menu do sistema, o usuário é direcionado primeiramente para a página de “Pesquisa de Projetos”. Nesta página, o usuário deverá clicar no botão “Novo” para que o sistema seja direcionado para a página de Cadastro de Projetos (Figura 5). Nesta página, o usuário deverá informar nome do projeto, sua descrição, a data de início e a data de fim do respectivo projeto. Após isso, ao clicar no botão “Cadastrar”, o usuário recebe a mensagem de “Cadastro efetuado com sucesso” e o sistema apresenta a opção de “Cadastrar Atividades” para o projeto recém cadastrado como mostra a Figura 6.



Figura 5: Cadastro de projetos

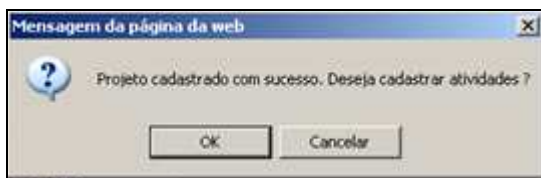


Figura 6: Mensagem de confirmação do cadastro de projeto

Vamos considerar que o usuário não deseje cadastrar atividades neste momento, então ao clicar na opção “Cancelar” o usuário será direcionado para a tela inicial do sistema.

Para cadastrar uma atividade, o gerente deve acessar o menu de “Atividades” na “Tela Inicial”. Ao clicar no item do menu “Atividades”, o gerente vai ser direcionado para a tela de “Pesquisa de Atividades”, onde deverá clicar na opção “Nova”, sendo assim direcionado para a página de “Cadastro de Atividades” (Figura 7), com todos os campos liberados para inserção da atividade.



Figura 7: Cadastro de atividades

Como pode ser visto na Figura 7, é possível cadastrar ao mesmo tempo diversas atividades em um projeto. Preenchendo os campos de cadastro e clicando em “Adicionar”, a atividade entra para uma lista na parte inferior da tela. Para finalizar o cadastro das atividades, o usuário clica em “Finalizar”. Caso queira editar ou excluir alguma das atividades cadastradas antes de finalizar, o gerente deve clicar nos respectivos botões, “Editar” e “Excluir”.

Ao confirmar a inclusão das atividades, o sistema disponibiliza a mesma função encontrada no Cadastro de Projetos. O sistema pergunta se o gerente deseja cadastrar tarefas para a atividade recém cadastrada (Figura 8). Assim como no cadastro de projeto visto anteriormente, vamos considerar que o gerente não deseje cadastrar tarefas neste momento, então ao clicar na opção “Cancelar” o gerente será direcionado para a “Tela Inicial” do sistema.

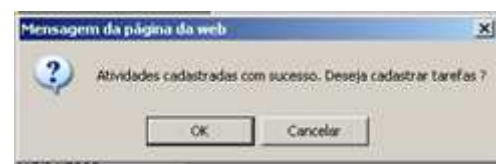


Figura 8: Mensagem de confirmação do cadastro de atividade

Para cadastrar uma competência no sistema, o gerente deve acessar o menu “Competências” da “Tela Inicial” do sistema. Na tela de “Pesquisa de Competências” que é aberta, o gerente deverá selecionar a opção “Nova” e é então direcionado para a tela de “Cadastro de Competências”. Nessa tela, como mostra a Figura 9, o gerente informa o nome da competência e em seguida clica em “Cadastrar” para confirmar a inserção da competência.





Figura 9: Cadastro de competência

Já que demonstramos como é realizado o cadastramento de uma competência, podemos agora passar ao cadastramento de papel, pois veremos a seguir que o cadastro de um papel está condicionado à existência de competências previamente cadastradas.

Para cadastrar um novo papel no sistema, o gerente deve acessar o menu “Papel” na “Tela Inicial” do sistema. A tela de Pesquisa de Papeis será exibida e o usuário deverá selecionar a opção “Novo”. Logo após o gerente é direcionado para a tela de Cadastro de Papéis, como mostra a Figura 10. Nesta tela, o gerente informa o nome do papel e inclui as competências que dizem respeito ao papel que esta sendo cadastrado. Em seguida o gerente deverá clicar em “Cadastrar” para confirmar a inserção do papel.



Figura 10: Cadastro de papéis

Uma vez cadastradas as competências e os papéis necessários, podemos realizar o cadastramento das tarefas do projeto.

Ao acessar o item de menu “Tarefas” da “Tela Inicial” do sistema, o gerente será direcionado para a tela de “Pesquisa de Tarefas”, onde deverá clicar na opção “Nova”, sendo assim direcionado para a tela de

“Cadastro de Tarefas” (Figura 11), com todos os campos liberados para inserção da tarefa.

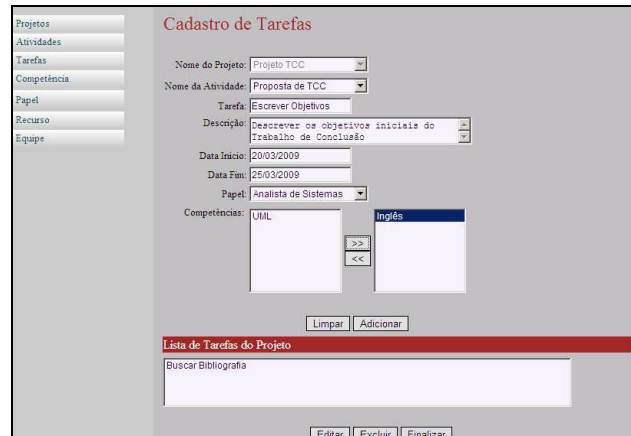


Figura 11: Cadastro de tarefas

Como pode ser visto na Figura 11, é possível cadastrar ao mesmo tempo diversas tarefas em uma atividade preenchendo os campos de cadastro e clicando em “Adicionar”. Para finalizar o cadastro das tarefas, o gerente clica em “Finalizar”.

Para cadastrar um novo recurso no sistema, o gerente deverá acessar o menu “Recurso” da “Tela Inicial” do sistema. Na tela de “Pesquisa de Recursos” que é aberta, o gerente seleciona a opção “Novo”. Logo após, o gerente é direcionado para a tela de “Cadastro de Recurso”, como mostra a Figura 12. Então, o gerente informa o nome do recurso, seleciona o papel que o recurso vai executar e inclui as competências que o recurso possui. Em seguida deverá clicar em “Cadastrar” para confirmar a inserção do recurso.



Figura 12: Cadastro de recursos

Para cadastrar uma nova equipe no sistema, o gerente deve acessar o menu “Equipe” na “Tela Inicial” do sistema. Na tela de “Pesquisa de Equipe”



que é aberta, o gerente clica no botão “Nova” e o sistema exibe a tela de “Cadastro de Equipe”, como mostra a Figura 13. O gerente, então, informa um nome para identificar a equipe e seleciona os recursos que farão parte da equipe. Na listagem dos recursos disponíveis, são listados somente os recursos que não estão em nenhuma equipe. Em seguida o gerente clica em “Cadastrar” para confirmar a inserção da equipe.



Figura 13: Cadastro de equipes

A alocação de recursos é realizada na tela de “Visualização” ou “Edição de Atividades”. O gerente pesquisa pela atividade que deseja fazer a alocação e clica na opção de “Visualizar Atividade”. Se a atividade estiver corretamente cadastrada, o botão de “Alocar Equipe” estará disponível (Figura 14). Para realizar a alocação então, basta o usuário clicar neste botão e logo após o sistema informará qual equipe foi alocada para realizar a atividade (Figura 15).



Figura 14: Visualização de atividade para alocação de equipe



Figura 15: Identificação de equipe alocada

Nesta seção foi apresentado o fluxo principal, necessário para realizar a alocação de uma equipe e em um projeto. Após todos os recursos e equipes serem cadastrados no sistema, o gerente de projetos irá cadastrar os novos projetos e atividades que serão desenvolvidos futuramente, utilizando o sistema para auxiliá-lo a realizar a alocação automatizada dos recursos.

#### 4. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Projetos com equipes geograficamente dispersas tem-se tornado cada vez mais comuns atualmente, e com esta tendência temos uma série de novos problemas que estão relacionados a este tipo de projeto. Um desses problemas é, em muitos casos, o desconhecimento da equipe pelo gerente de projetos, suas competências, habilidades, entre outros, que impedem as tomadas de decisões do gerente em relação à equipe mais adequada, e que tenha disponibilidade para a realização de uma nova atividade a ser realizada no projeto.

O sistema WEBSMIT, tem como objetivo auxiliar o gerente de projetos identificando a disponibilidade e as competências das equipes para realização de atividades em projetos e realizando a alocação da melhor equipe disponível para realizar a atividade necessária ao projeto.

Durante a realização deste trabalho, nas pesquisas realizadas, não encontramos ferramentas que realizem a alocação automática de recursos para equipes de projetos geograficamente distribuídas.

Atualmente estamos trabalhando na validação da funcionalidade do sistema. Uma validação inicial será

feita para se comparar a realização da alocação de recursos da maneira tradicional, feita por um gerente de projetos com conhecimento das suas equipes com a alocação automática realizada pelo sistema WEBSMIT.

A versão atual do sistema WEBSMIT foi desenvolvida em C# com banco de dados SQL Server 2005 Express. A escolha por esta linguagem de desenvolvimento ocorreu devido ao trabalho ter sido desenvolvido no âmbito de um trabalho de conclusão de curso que possui um tempo de execução limitado e pelos alunos envolvidos no projeto já possuírem experiência de programação nesta linguagem.

Como trabalhos futuros pretendemos realizar a implementação do sistema utilizando ferramentas específicas para desenvolvimento de agentes e realizar uma comparação de performance entre as duas versões do sistema. Outros trabalhos futuros a serem realizados são a descentralização da base de dados e a possibilidade de se alocar algum ou alguns dos recursos de uma equipe para a realização de uma tarefa ao invés de se alocar a equipe como um todo.

#### REFERENCES

- [1] S. Krishna, S. Sahay, G. Walsham. Managing Cross-Cultural Issues in Global Software Outsourcing. *Communications of the ACM*, New York, 47(4): 62-66, 2004.
- [2] Project Management Institute. A guide to the project management body of knowledge (PMBOK guide). Project Management Institute. Pennsylvania, 2004.
- [3] J. Audy, R. Prikladnicki. Desenvolvimento Distribuído de Software. Campus: Rio de Janeiro, 2008.
- [4] G. Weiss. Multiagent Systems - A modern Approach to Distributed Artificial Intelligence. The MIT Press: Cambridge, Massachusetts, 1999.
- [5] J. Bradshaw. Software Agents. AAAI Press/The MIT Press: Menlo Park, California, 1997.
- [6] L. O. Alvares, J. Sichman. Introdução aos Sistemas Multiagentes. In *Jornada de Atualização em Informática, Congresso da SBC, 17.*, SBC:Brasília, p.1-38, 1997.
- [7] S. Russell. Analogy by Similarity. In David Helman (Ed.), *Analogical Reasoning*, D. Reidel, Boston, 1988.