

PROTOTIPO DE UM SISTEMA DE RECOMENDAÇÃO NA GESTÃO DE COMPETÊNCIAS EM DISCIPLINAS DA ÁREA ACADÊMICA

SPEROTTO, Fabio Aiub

TONI, José Alexandre

Resumo

Os sistemas de informação se tornam mais complexos e especialistas a cada dia, reunindo dos mais variados conceitos ao desenvolvimento de software que atendam a uma demanda crescente de ferramentas de gestão. O presente artigo visa trazer conceitos da ciência e a sua aplicação no desenvolvimento de programas, mais especificamente na construção de sistemas de recomendação. Os conceitos apresentados serão: introdução aos aspectos da gestão; a questão das competências dos professores; a usabilidade da Ontologia entre definições de conhecimento; a inserção da mineração textual na técnica de comparação de características; como conceitos de inteligência artificial em lógica nebulosa podem ser inseridos em construção de software e o que são os chamados sistemas de recomendação. Após a enumeração destes conceitos, a construção do protótipo em suas fases importantes é apresentada.

Palavras-chave: ontologia, gestão, mineração, conhecimento, sistema.

1. Introdução

Sistemas de informação englobam uma gama de softwares para os mais variados fins, sendo um deles de grande interesse, as ferramentas de apoio a gestão. Estas ferramentas podem ou não conter especialidade na área alvo ou as vezes até mesmo trazer mais obstáculos do que ajudar no apoio aos gestores. Sistemas de recomendação trazem a novidade da qual as ferramentas iniciam sua participação na tomada de decisão pelo seu usuário e desta forma começa, a ser mais comuns entre as organizações a nível gerencial.

As seções seguintes demonstraram conceitos de práticas gerenciais (seção 2 a 3), alguns conceitos básicos sobre ontologia (seção 4), lógica nebulosa (seção 5) e em seguida conceitos sobre sistemas de recomendação (seção 6). Exibindo os estudos sobre a proposta de um protótipo que visa apoiar a tomada de decisão pelos gestores do centro acadêmico. Sendo o centro acadêmico escolhido como objeto de estudos a Unochapecó.

2. Gestão Universitária

As questões que envolvem um sistema voltado para uma instituição vão além dos parâmetros normais da tecnologia empregada e dos requisitos puramente técnicos. Faz-se necessário, na abordagem de um sistema voltado aos gestores de áreas da universidade, compreender como funciona a estrutura da organização educacional.

A organização educacional é um agrupamento humano em interação, que, ao relacionar entre si, e com o meio externo, faz uma construção social na realidade, propiciando seu ciclo de sobrevivência [...] (TACHIZAWA, 1999). A partir desta reflexão pode-se compreender a responsabilidade dos gestores da unidade acadêmica, em gerenciar o fluxo de informações e o conhecimento.

Para este gerenciamento, o ambiente interno da organização é o melhor ambiente de fácil expansão e de melhoria em sua qualidade (COELHO, 2008). Neste ambiente, os gestores podem conceber mecanismos de melhoria dos seus profissionais, já que é um ambiente mais próximo da atuação gerencial. Desta forma, o emprego de estratégias, segundo ainda Tachizawa (1999), podem ser úteis aos planos de gestão:

- Melhoria da qualidade das IES como um todo, aumentando o rigor com relação ao corpo docente e quanto aos cursos oferecidos aos clientes (alunos).
- Utilização das tecnologias da informação para o gerenciamento do conhecimento e um melhor suporte administrativo.
- Criação e manutenção de um banco de dados de talentos, dos professores, devido a exigência de mão-de-obra altamente qualificada.
- Implementação de cursos de especialização, cursos sequenciais e outras formas de ensino complementares aos cursos de graduação, com estreita interação teoria-prática.

É de fácil visualização como estas estratégias podem ser inseridas em planejamentos ou ações de melhorias, inclusive, deixando claro que a tecnologia da informação precisa receber um foco em união a estes anseios a fim de conceber uma estratégia coerente com os objetivos da universidade.

3. Recursos Humanos e as competências

Gestão de Pessoas é a função gerencial que visa à cooperação das pessoas que atuam nas organizações para o alcance dos objetivos tanto organizacionais quanto individuais (MIGLIORI, 2006). Os objetivos específicos da organização, neste caso, remetem-se as estratégias específicas de cada instituição de ensino. Há uma significativa interligação entre essas administrações, justamente pelo fato dessa conjuntura entre estratégias e a ação dos funcionários como um todo. Pois todos estão reunidos em prol do desenvolvimento da organização e em busca da efetivação da missão da instituição.

Apesar da gestão ser abrangente a todos os envolvidos na instituição, nesta trabalha o ator principal é o professor. Para que a qualidade comentada anteriormente se perpetue, uma melhor organização por parte dos docentes pode ser de grande valia, especialmente para as disciplinas, cursos e alunos de qualquer instituição.

Os cargos devem ser projetados para apoiar as etapas do processo, bem como os ambientes do cargo devem ser estruturados para permitir que as pessoas dêem sua contribuição máxima à eficácia e à eficiência organizacionais (TACHIZAWA, 1999). Essa análise comentada pelo autor, não necessariamente seria, portanto, somente com os professores. Mas sim com todos os envolvidos, na busca incessante de melhorias do ambiente de trabalho e dos profissionais nos seus postos de trabalho.

Para o professor isso é ainda mais interessante, pois além da sua capacidade profissional descrita no seu currículo, outras variáveis dependem do sucesso da eficiência do seu ensino. Ter um gestor que possua a capacidade de definir por meio de ementa e planos pedagógicos quais, entre professores aptos, aqueles que poderiam entregar o melhor ensino, entre as exigências quem imperam na instituição. Um sistema informacional pode realizar um suporte administrativo sobre esta responsabilidade de gerenciamento.

O desenvolvimento da organização está diretamente relacionado com a sua capacidade em desenvolver pessoas e ser desenvolvida por pessoas [...] (DUTRA, 2002). Esta definição entre em união com a própria estratégia universitária, que pode ser a própria melhoria da instituição de ensino. Esta melhoria pode ter seu início da visualização das competências do

próprio professor. A competência em si pode ser considerada, segundo Dutra (2002), um *cluster* de conhecimentos, habilidades e atitudes relacionados que afetam a maior parte de um papel profissional ou responsabilidade, que se correlaciona com a desempenho desse papel ou responsabilidade, que possa ser medido contra parâmetros “bem aceitos”.

Como o objeto de estudo é a Unochapecó, foi investigado a respeito destes parâmetros, em entrevistas com profissionais da área de gestão de pessoas. As técnicas mais usadas para visualizar os atributos dos professores partem de avaliações por meio de entrevistas, provas escritas e pontuação sobre as características já existentes dos candidatos a professores.

Estas avaliações são realizadas de forma prescritiva, sem adoção de um sistema de informação pleno, como suporte a tomada de decisão na escolha, do considerado melhor profissional, para uma determinada disciplina. Nas seções seguintes são descritas quais os atributos de competências foram definidos e quais são os parâmetros considerados e aceitos pela organização em estudo, entre a escolha de um grupo de professores para uma disciplina.

4. Ontologia

Ontologia pode ser uma maneira de se conceitualizar de forma explícita e formal os conceitos e restrições relacionados a um domínio de interesse (Carneiro et al. apud Guarino 1998). Gruber (1992) define a ontologia como um vocabulário de termos (com funções, relacionamentos) em que seres humanos e máquinas possam ler e que sistemas baseados em conhecimento possam interoperar no nível de conhecimento. Para a ciência da informação, ontologia é a representação da informação em meio digital, relacionando-se com conteúdos da realidade (Gonçalves et al. 2008).

Ontologia é um conceito que permeia, portanto, um mecanismo que formaliza um domínio de informação (um conhecimento específico). Definem de forma clara e objetiva os conceitos que fazem parte de determinado domínio e que precisam ser formalizados para que tanto as pessoas possam compreender, mesmo com diferentes opiniões, quanto as máquinas que possam reusar informação em seu nível abstrato.

No protótipo em questão, este conceito foi aplicado como ferramenta de mapeamento das características do ambiente da qual o sistema é aplicado. Reúne os conceitos e relacionamentos de todo o domínio da aplicação, formalizando assim o que precisa ser considerado para a estrutura lógica do *software*.

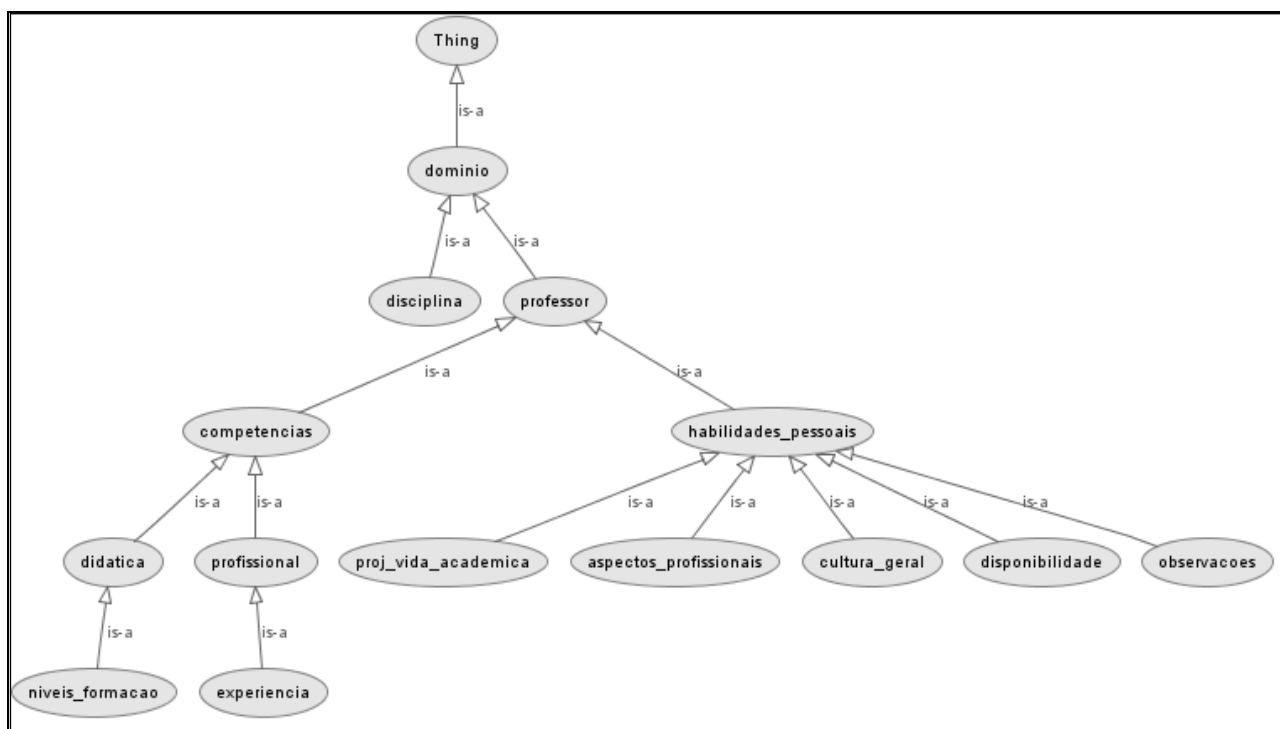


Figura 1. Mapeamento ontológico.

Para o mapeamento visualizado na figura 1, foi escolhido o programa Protégé¹. Esta ferramenta disponibiliza uma estrutura para que se possam desenvolver modelos ontológicos, e estes, podem ser exportados em vários tipos de documentos, seja em imagens ou em documentos para leitura por máquinas.

Nesta figura 1 pode ser visto uma visualização da combinação de classes e subclasses. Geralmente a ontologia descreve classes como sendo os próprios conceitos. Propriedades de dados são os tipos de dados que são inseridos para as classes. Assim, foram classificados os seguintes conceitos:

- Thing: classe raiz, contém todas as classificações de projeto, faz parte tanto da ferramenta ontológica quanto das documentações da w3c² para a ontologia, significa simplesmente “alguma coisa”.
- Dominio: o domínio de interesse da ontologia, neste caso refere-se à Unochapecó, mais especificamente centro acadêmico. A escolha da palavra Domínio é apenas uma referência ao domínio da qual o presente protótipo se refere ontologicamente, englobando as disciplinas e os professores.
- Disciplina: no conceito de disciplina, existente normalmente, dois tipos de dados foram classificados internamente: título e ementa.
- Professor: maior conceito dentro do domínio. Define o professor e suas subclasses (conceitos relacionados), sendo os conceitos relacionados às Competências e Habilidades Pessoais. O único tipo de dado concedido a ele é um “XML Literal”, pois todos os dados são importados de um arquivo XML que, por sua vez, é exportado da plataforma Lattes.
- Competências: dividem-se em Didáticas e Profissionais. A didática se refere às graduações do indivíduo (a aplicação trata desde graduação até pós-doutorado). As Profissionais são referentes às atuações profissionais encontradas no currículo do

1 <http://protege.stanford.edu>

2 <http://www.w3.org>

indivíduo, sendo que a norma existente verifica os anos de experiência independente da origem do trabalho.

- Habilidades pessoais: divide-se em conceitos como projeto de vivência acadêmica, aspectos profissionais, cultura geral, disponibilidade de tempo e observações. Estes conceitos foram encontrados em uma relação entre entrevistado e entrevistador. Descrições textuais definem cada grupo destes conceitos, sendo, portanto definido interiormente um tipo de dado como texto.

A ontologia forneceu mecanismos com os quais foi viabilizada a modelagem do conhecimento seguindo o modelo organizacional vigente. Forneceu não somente entendimentos para a aplicação, mas também para toda a gestão organizacional, como os profissionais definem o conhecimento e alguns dos seus relacionamentos, formalizando aquilo que muitas vezes é crença ou se diz de forma textual sem a centralização de registros.

5. Lógica Nebulosa

Os conceitos que definem o que é inteligência artificial (onde a categoria de lógica nebulosa se insere) também divergem entre os autores na literatura. Muitos preferem não conceituar esse termo e partir diretamente para modelos computacionais que possam ilustrar essa inteligência.

Basicamente, a inteligência artificial pode assim ser definida: o estudo de como fazer os computadores realizarem tarefas que, no momento, as pessoas são melhores (Rich, 1988). É o uso do computador para executar raciocínio, reconhecimento de padrões, aprendizagem ou outras formas de inferência (Luger, 2004). O objetivo central da IA (Inteligência Artificial) é simultaneamente teórico – a criação de teorias e modelos para a capacidade cognitiva – e prático – a implementação de sistemas computacionais baseados nestes modelos (Bittencourt, 2001). Dessa forma visualizamos um novo campo de desenvolvimento de sistemas, cada vez mais inteligentes e fornecedores de conhecimento, até mesmo na postulação de ações e de novo conhecimento, podendo tornarem-se úteis a indivíduos tomadores de decisão.

Contemporaneamente, podemos encontrar sistemas especialistas que ajudam a promover a intenção desses conceitos comentados no parágrafo anterior. Esses sistemas são chamados de sistemas especialistas e possuem geralmente três arquiteturas: uma base de regras, uma memória de trabalho e um motor de inferência (Bittencourt, 2001). A base de regras constitui do “alimento” do funcionamento do sistema, é onde perguntas serão feitas, condições arquitetadas serão promovidas para que o sistema possa gerar um conhecimento. A memória de trabalho também participa dessa geração e pode ser qualquer estrutura de dados. O motor de inferência é que usará os subsídios das duas partes anteriores: que irá aplicar tanto as regras quanto as informações disponíveis na memória, para fazer as devidas inferências que serão retornadas ao usuário.

O maior problema visualizado é a determinação da inferência sobre um determinado conjunto de informações. O sistema receberia as ideias e iria inferir sobre elas, determinando como resultado o que ele poderia “mensurar”, “quantificar” ou “pensar” sobre aquele domínio de problema. Existem várias técnicas reconhecidas na literatura de IA, contudo este trabalho, respondendo ao problema citado, é definido um pequeno estudo sobre a lógica *fuzzy*.

A lógica *fuzzy* faz parte do estudo sobre lógica nebulosa, que é o modelo mais tradicional para o tratamento da informação imprecisa e vaga (Bittencourt, 2001). A lógica *fuzzy* foi introduzida no contexto científico em 1965 pelo professor Lotfi Zadeh e se diferencia da lógica booleana, 0 e 1, precisão de resposta (JANÉ, 2004). Essa lógica têm como ponto

fundamental a representação da lógica e da racionalidade humana na resolução de problemas complexos (Bittencourt apud Von Altrock, 2004).

Em sistemas comuns computacionais a lógica booleana é usada, o computador processa uma informação e define se aquilo é correto ou não, precisamente. Entretanto, no trato com sistemas mais inteligentes que operam com informações do tipo “hoje o clima está mais ou menos quente”, ou “aquele indivíduo é mais ou menos competente em uma tarefa”, o conjunto de dados dá-se de forma diferente. Um determinado valor têm um determinado grau de pertinência em um conjunto, ainda usando o exemplo anterior, podendo hoje estar 40% frio e 60% quente (então está mais ou menos quente, dependendo do conjunto de regras estipulado para o entendimento pelo sistema).

Por meio do estudo na literatura a forma mais contemplada de resolver o problema da ferramenta no enfoque de gestão de competências seria a lógica *fuzzy* (lógica nebulosa). Pois a indução de regras no modelo de avaliação de competência se faz necessário existir (como abordado nos capítulos anteriores). E os indivíduos (professores) também seriam avaliados, conforme a arquitetura, na questão de serem mais pontuados para uma determinado conjunto de valores (seriam mais propícios para uma determinada disciplina).

6. Sistemas de Recomendação

Sistemas de recomendação (SR's) surgiram como um conceito de aumentar o nível de respostas dos sistemas comuns. Aplicativos estes que na maioria das vezes fazem com que os usuários precisem vasculhar resultados estáticos a respeito dos dados de entrada. O objetivo dos Sistemas de Recomendação é o de que os usuários não apenas recebam o retorno de itens a partir da formulação de consultas, mas que o possível interesse por um determinado item possa ser previsto (Lichtnow et al., 2006). Burke ainda define que sistemas de recomendação modernos são sistemas que produzem recomendações individualizadas como saída ou possuem efeito de mostrar um caminho personalizado direto no objeto de interesse em meio a uma grande variedade de opções.

Esses sistemas são amplamente usados em *sites* de comércio (*e-commerce*). Segundo Schafer (et al. 1999), aplicativos são usados para sugerir produtos baseados naqueles que estão sendo mais vendidos ou, ainda, baseados no passado de compras de cada cliente. Alguns outros exemplos são o sistema de recomendação de livros da *Amazon*³, anunciante de cd's *CD Now*⁴ e também na sugestão de roupas, no caso de um aplicativo apresentado pela marca de roupas *Levis*⁵.

Um modelo de desenvolvimento de sistemas desse tipo, considerado um dos mais completos (e usados em alguns dos aplicativos citados acima) se define em quatro processos: identificação do usuário, coleta de informações, estratégias de recomendação e visualização das recomendações (Barcellos et al. apud Schafer, 2007). A identificação do usuário não é obrigatória desses sistemas, se existir, é o primeiro passo a ser executado. Características do usuário são reconhecidas e o sistema pode atender de forma personalizada para cada tipo de usuário (efeito de caminho personalizado comentado anteriormente).

Em seguida é feita a coleta de informações, nessa fase as informações são retiradas na busca de dados. Essa busca de dados pode ser feita por meio de dados inseridos pelos usuários, histórico de navegação, compras e demais ações passadas (assim como bases de dados). Também pode ser por meio de inferências feitas pelas características de um grupo de usuários

3 <http://www.amazon.com>

4 <http://www.cdnow.com>

5 <http://www.levis.com>

para um determinado usuário (comparação de preferências, gostos, atividades, entre outros). Essas informações podem ser reunidas em um banco de dados em que podem ser aplicadas técnicas de mineração de dados a fim de descobrir relações entre os dados obtidos e as informações desejadas.

Existem várias técnicas elaboradas ou agregadas que foram usadas para se desenvolver sistemas de recomendação. Algumas arquiteturas são catalogadas e propostas na literatura, todas baseiam-se em propriedades distintas tendo como mesmo objetivo o conceito de recomendação. As técnicas especificamente atendem a três requisitos básicos: os dados existentes no plano de fundo, antes do sistema de recomendação gerar novos dados; os dados de entrada, as informações que os usuários podem comunicar ao sistema para gerar a recomendação; e em um terceiro nível, os algoritmos que vão fazer as combinações entre os dados existentes e os dados de entrada para obter sugestões. Estes são os processos básicos e por meio destas etapas podem existir cinco grandes técnicas de desenvolvimento de SR: recomendação colaborativa, recomendação baseada em conteúdo, recomendação demográfica, recomendação baseada em utilização e recomendação baseada em conhecimento (Burke, 2002).

As técnicas de recomendação baseadas em utilização e conhecimento não são focadas em generalizar ou avaliar os usuários, suas preferências e seus históricos a longo prazo. A ideia central é sugerir um caminho entre a necessidade do usuário e um conjunto de opções disponíveis. No caso do SR baseado em utilização, a arquitetura parte da premissa de avaliar a utilidade de cada objeto para o usuário. Já o baseado em conhecimento consiste na sugestão baseada na inferência do que o usuário necessita e das suas preferências. Em outras palavras, para construir um sistema nesse modelo, seria necessário uma representação mais detalhada das necessidades do usuário (Bunker 2002 apud Towle & Quinn, 2000) para então suportar como um item em particular encontra uma necessidade particular.

Para o projeto, a escolha mais considerável é o baseado em conhecimento, pois de um lado existe a ementa das disciplinas, com toda a descrição e as necessidades do desenvolvimento da matéria. De outro lado, têm-se as características dos professores assim como a sua formação acadêmica. Há, por conseguinte, opções de um lado que precisam ser corretamente conectadas nas opções do outro lado. Um exemplo é se a ementa se refere à matéria que ensina tópicos de programação para a *web*, temos umas opções que podem ser a linguagem em *script*⁶ e estilização de páginas. E temos alguns professores que ensinam linguagem em *script* e outros que entendam o básico de estilização. A necessidade do gestor seria de uma sugestão de que professor pudesse preencher as opções da respectiva ementa. Essa técnica (baseada em conhecimento) torna-se a mais próxima da arquitetura necessária a ser desenvolvida no protótipo. Já que nas demais, são focadas nas inferências de desejos de vários usuários associados a várias características e avaliações diretas sobre objetos e não sobre a necessidade de definição de que “conhecimento poderá suprir conhecimento” (qual o professor ideal para ministrar determinada disciplina). As características dos usuários, como descrito, não é obrigatória, não fará parte do desenvolvimento pois não há uma necessidade de tratar os atributos pessoais dos usuários do sistema. A coleta de dados será por meio de informações inseridas pelo gestor, e banco de dados formado dos registros das ementas de disciplinas e formações acadêmicas dos professores.

7. Desenvolvimento do protótipo

O desenvolvimento da proposta se dará como um sistema voltado para a *web*. Dessa forma

6 Linguagens de programação que são interpretadas. Um navegador *web*, pode interpretar a estilização de cores e estruturas que são programadas para uma determinada página, por exemplo.

algumas tecnologias devem ser consideradas para o desenvolvimento pleno do *software*. Questões como arquitetura, escalabilidade e boas práticas devem ser observadas para desenvolvimento de qualidade de qualquer aplicativo para a *web*. Na parte de programação foi escolhida a linguagem PHP, pois poderá atender perfeitamente as questões anteriores de ótimo desenvolvimento para a *web*.

Conforme o desenvolvimento, alguns itens precisarão ser considerados como a interação com usuário e, dessa forma, pode ou não surgir a necessidade do uso de *API's* (classes de códigos automatizados feito por terceiros, como uso de Javascript (Jquery⁷ se torna interessante pois é uma biblioteca que automatiza o desenvolvimento de uma melhor interação da interface). O uso do componente *ajax*⁸ do Jquery é usado para as telas exibidas ao usuário final. XHTML e CSS também serão usados, pois o primeiro, define a estrutura de qualquer página e o segundo define o estilo que ela conterá. Estas tecnologias de desenvolvimento de páginas serão codificadas dentro de padrões designados pela W3C (www.w3.org). O banco de dados escolhido para a pertinência dos dados é o MySQL⁹.

A codificação é realizada por meio da linguagem de programação PHP¹⁰, utilizando as melhores práticas de desenvolvimento web. Nestas práticas, é usado como *framework* para linguagem, o CodeIgniter¹¹. Este *framework* foi escolhido pela sua ótima performance, facilidade de aprendizado, tem o foco em soluções simples e por possuir boa documentação.

A figura 2 exibe a disposição da tela inicial, três grandes funções do sistema são

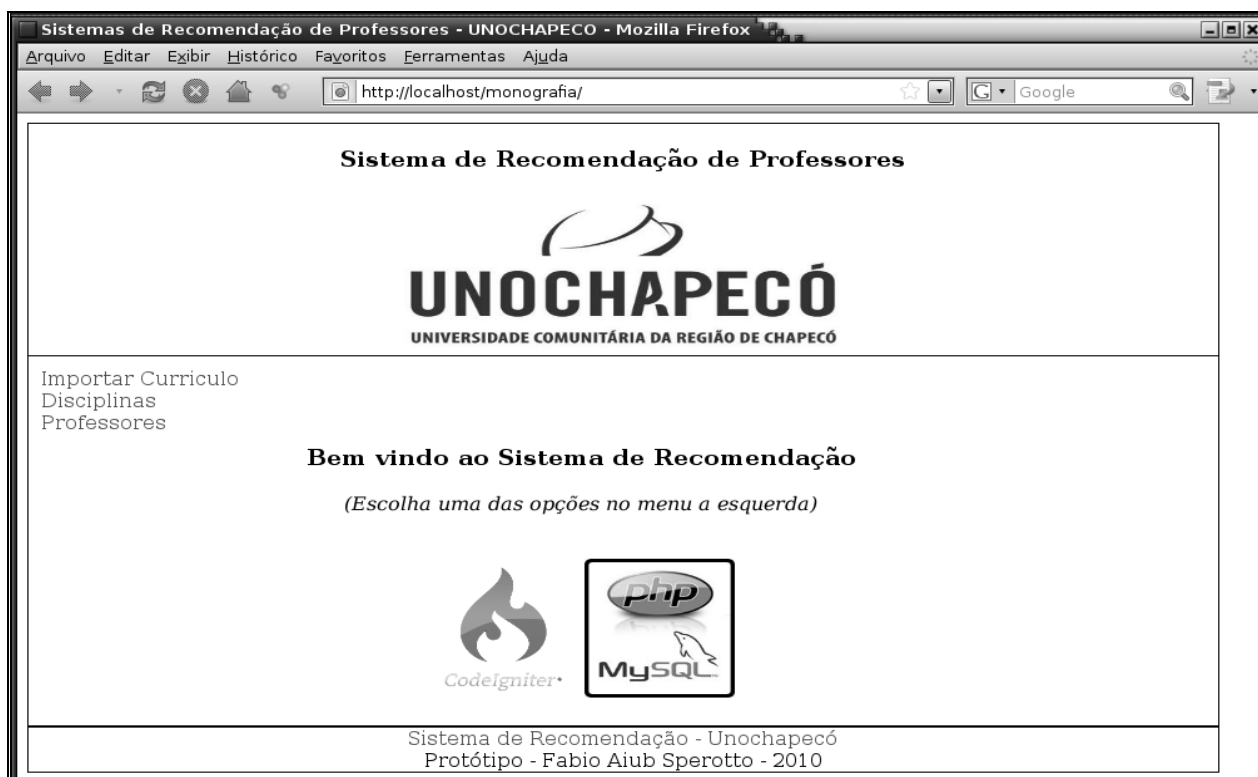


Figura 2. Tela inicial do sistema.

7 <http://www.jquery.com/>

8 <http://api.jquery.com/category/ajax/>

9 <http://www.mysql.com/>

10 <http://www.php.net/>

11 <http://www.codeigniter.com>

exibidos no *menu* a esquerda: Importar Currículo, Disciplinas, Professores. A importação de currículo é considerado um dos pontos fortes do sistema. Como os currículos dos professores são dispostos na plataforma LATTES¹², que é do próprio Ministério da Educação. Todos os currículos seguem um mesmo padrão e estes podem ser exportados via documentos XML. Desta forma não houve a necessidade de criar vários formulários de cadastros de professores. Com o componente do PHP, o SimpleXML¹³, é realizado a importação ou atualização de dados pessoais, níveis de formação e atuações profissionais dos professores em poucos segundos.

Na opção Disciplinas, são cadastradas as ementas das disciplinas da instituição, de forma textual. Também para este protótipo é mostrado uma listagem automática de todas as disciplinas já cadastradas no sistema.

Na opção Professores, podem ser pesquisados todos ou um professor específico. Nesta seção do sistema podem ser cadastradas as habilidades pessoais, oriundas de alguma entrevista entre gestor e candidato a professor. Também pode ser realizado um controle de disciplinas da qual o professor já seja vinculado, entretanto, não é obrigatório para o funcionamento do sistema de recomendação. Para que o sistema realize uma recomendação, inicialmente deve ser escolhido a opção Disciplinas e nesta escolher uma disciplina específica (para o estudo, logo abaixo, é escolhida uma disciplina de Algoritmos), assim que for realizado estas operações, aparece um tela como visualizado na figura, a seguir.

Recursos de inferência

Disciplina: Algoritmo e Estrutura de Dados I

Configure abaixo os pesos

*Tem como objetivo acentuar o que é mais necessário.
Para que o motor de inferência possa se ajustar na sugestão mais desejada pelo gestor*

Competências Didáticas:

Competências Profissionais:

Habilidades Pessoais:

Professores com a disciplina marcada?

Figura 3. Painel de configuração da recomendação.

Com este painel de controle, são fornecidos ao gestor alguns itens que podem ser configurados. O objetivo é que o próprio usuário possa parametrizar o tipo de sugestão

12 <http://lattes.cnpq.br/>

13 http://www.php.net/manual/pt_BR/book.simplexml.php

mediante pesos. Os três conjuntos de características definidas na ontologia (competências didáticas, profissionais e habilidades pessoais) possuem um peso padrão considerado baixo. Entretanto podem ser configurados entre as opções “Nenhum” a “Alto” (0 a 3 respectivamente), fazendo com que se mude a pontuação da relevância na sugestão. Outra opção é com relação a disciplina escolhida: caso a resposta descrita na tela a respeito da marcação de disciplina for positiva, a sugestão será fornecida baseada somente nos professores que foram previamente cadastrados para esta disciplina (vinculados a disciplina, por meio do no menu Professores). Após clicar no botão Enviar, o gestor receberá uma resposta, logo abaixo do painel, parecida com o que se pode observar na próxima ilustração.

Professor Aptidão	
Professor1	53.9%
Professora1	24.6%

Figura 4. Resposta do sistema de recomendação (formato de ranking).

Nesta figura 4, levando em consideração dois professores previamente cadastrados, é dito que o Professor1 tem mais aptidão para ministrar a disciplina que a Professora1. Desta forma, independente da quantidade de professoras que o sistema possa sugerir, sempre mostrará no topo da lista o mais apto para a disciplina escolhida.

8. Estrutura lógica do sistema

A codificação foi desenvolvida em paralelo com as interfaces apresentadas anteriormente. Não houve um levantamento de requisitos realizado de forma mais apurada pelo fato da reunião de conceitos diferenciados que foram trazidos para este projeto. A maior parte do tempo entre desenvolvedor e cliente foi para o entendimento da inovação que o sistema poderia trazer. Após a ontologia formada, o primeiro passo foi a realização da modelagem entidade-relacionamento e, por conseguinte, é descrito os principais algoritmos desenvolvidos.

8.1 Modelagem do Banco de Dados

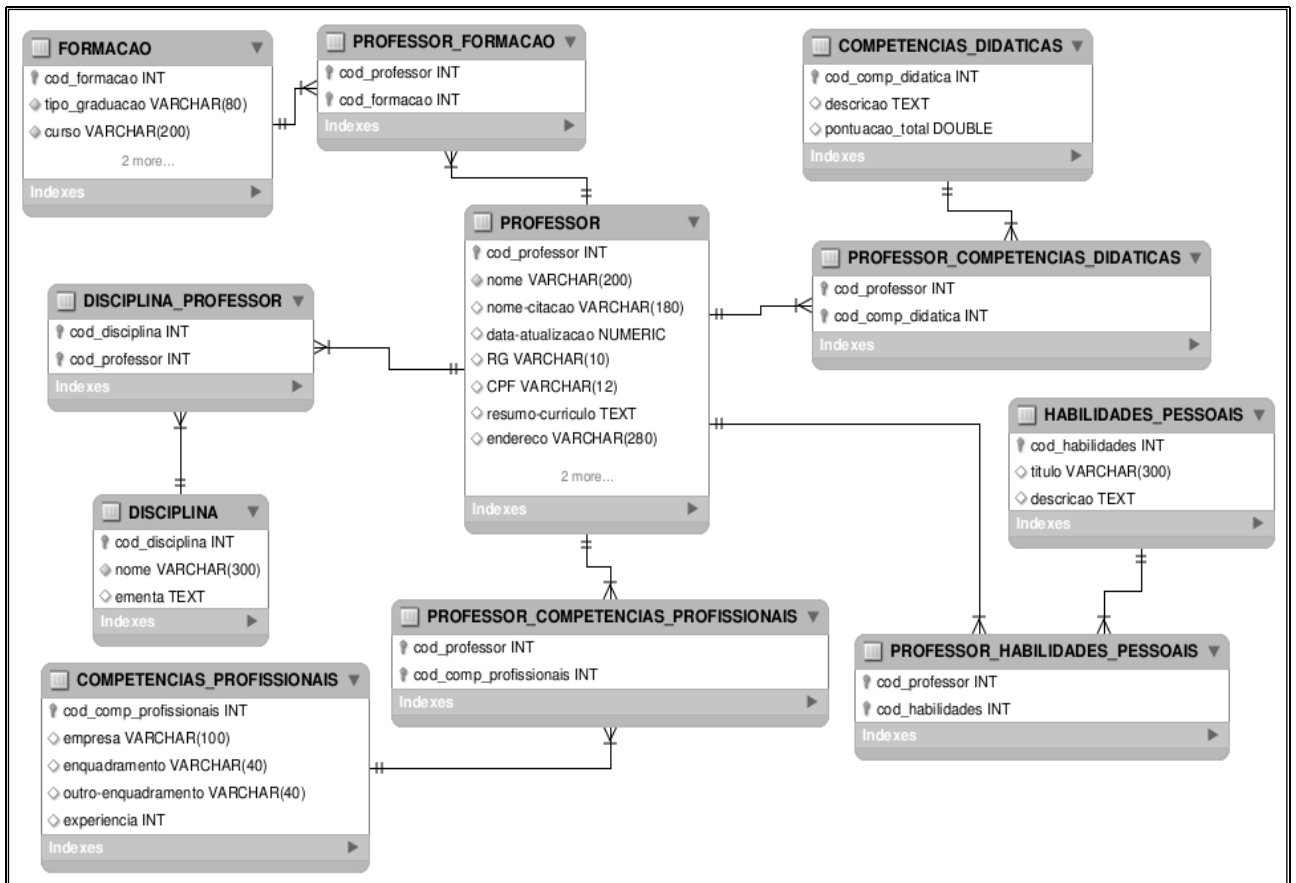


Figura 5. Modelagem entidade-relacionamento do sistema.

Segue a disposição da modelagem na figura 5. O cadastro de professores segue com as informações obtidas por mediante a importação de currículo e desta forma permanecem registradas na tabela PROFESSOR (algumas estruturas seguem a risca a nomenclatura que a plataforma LATTES usa nas nomeações de dados).

Os níveis de formação (que fazem parte também das competências didáticas) ficam à parte na estrutura FORMACAO e as atuações profissionais, como definidos na ontologia (seção 4) em COMPETENCIAS_PROFISSIONAIS. Na HABILIDADES_PESSOAIS permanecem os dados provenientes das entrevistas feitas hoje de forma digitada em documentos de textos, mas que agora possam ser registradas em banco de dados.

A existência da estrutura COMPETENCIAS_DIDATICAS é disponibilizada para reservar um local apropriado para futuros registros sobre estes aspectos relacionados ao perfil do professor. Na próxima seção, é discutida a codificação referente aos aspectos existentes no processo seletivo e o que o protótipo acolhe.

8.2 Interpretação e Pontuação Fuzzy

Nesta fase, levou-se em consideração o desenvolvimento contínuo da modelagem no banco de dados assim como os debates e acesso a materiais de seleção. A interpretação *fuzzy* foi desenvolvida levando em consideração o questionamento de como um algoritmo poderia ser desenvolvido que pudesse realizar a sugestão dos professores baseados em características

qualitativas e quantitativas. Desta forma como estudado na seção 2.4.3 a ideia é de usar a interpretação *fuzzy* para determinar um professor mais ou menos apto para ministrar uma disciplina e quanto mais apto é com relação aos demais professores.

As características reconhecidas quantitativas são as pontuações existentes no processo seletivo, para os níveis de formação (graduação até pós-doutorado) e a atuação profissional. Outros itens foram observados, mas somente estes foram agrupados para o presente protótipo. Nas características qualitativas, houve esforço maior no desenvolvimento lógico, pois exige um cuidado diferenciado devido aos dados serem em formato textual.

Um material usado como suporte na modelagem do algoritmo é de título “Controle e Modelagem *Fuzzy*”¹⁴. Especificamente por meio de análise da modelagem com variáveis linguísticas (página 24), possuindo relação com as propriedades fundamentais de conjunto (página 35 possui conexão conceitual sobre conjuntos disjuntos interpretados na ontologia, seção 4) e algumas relações *fuzzy* como base de conhecimento (a partir da página 42).

Na terminologia de sistemas inteligentes, a relação *fuzzy* é uma base de conhecimento, ou seja, o depósito de toda a inteligência relacionada a um dado sistema. Assim como a função de transferência na teoria de controle linear, a relação *fuzzy*, se conhecida, nos permite computar a resposta do sistema a uma dada excitação. A diferença entre as duas é que relações *fuzzy* não exige requisitos de linearidade ou de invariância temporal [...] (Shaw 1999, pág. 43-44).

Assim, pelo fato de o sistema não possuir requisitos mais rígidos o autor ainda comenta que dois itens devem ser levados em consideração pelo projetista: como a identificação e estimação dos conjuntos relacionados ao domínio do problema. Sendo assim, reunindo o conhecimento alcançado nos estudos e pesquisas, definiu-se o desenvolvimento por meio de pontuações ou *ranking* a respeito do condicionamento do professor para a disciplina.

Os conjuntos foram definidos por meio de equações e pesos para formalizar os grupos de excitação (que respondem por meio da mudança dos seus pesos, pelo usuário, como mostrado na figura 3). Os pesos definidos pelo usuário são considerados na anotação Pu. Estes pesos vão de nenhum até alto, ou 0 até 3. Se for escolhido zero, já é definido a não execução da regra matemática.

O universo de discurso é definido por U sendo $U = \sum R_p$. R_p é o *ranking* de um professor específico, portanto universo de discurso é interpretado como a somatória dos *rankings* dos professores, e R (recomendação) é meu próprio universo mais filtros de interpretação ao usuário final: $R = U$. R_p é definido por $R_p = \sum C_d + \sum C_p + \sum H_p$. Sendo o total das somas entre os cálculos de competências didáticas (C_d), competências profissionais (C_p) e habilidades pessoais (H_p). Estas formam o escopo matemático do algoritmo. Os detalhes ligados diretamente ao perfil do professor são discutidos a seguir.

As habilidades pessoais são informações cadastradas em formato texto. Para que se possam viabilizar as pontuações referentes a estas descrições, foram concebidos mecanismos de mineração e contagem textual. Estes mecanismos realizam o confronto entre as descrições das habilidades pessoais, com o título e a ementa da disciplina alvo. Nesta forma, é feita uma combinação entre as palavras de cada lado da descrição e, em cada retorno positivo, soma-se um valor matemático, tornando-se assim uma forma quantitativa para a resolução matemática.

Os mecanismos que realizam a mineração textual foram codificados usando funções existentes da própria linguagem PHP. Estas funções podem ser encontradas no guia de

14 Autor Ian Shaw e tradução de Marcelo Godoy Simões.

referência da linguagem¹⁵. Foi considerado o problema de combinações de palavras, verbos, artigos, preposições, pontuações, inúteis ou que não fazem parte do objetivo da aplicação e podem retornar falsas combinações positivas.

As funções da linguagem usadas, são as seguintes:

- `str_replace`¹⁶: é usada para substituir palavras ou expressões que não podem ser consideradas para a comparação de textos (evita falsos positivos).
- `explode`¹⁷: da forma como foi realizada a aplicação de filtro por meio de `str_replace`, `explode` é usado para que transforme o texto em um *array* de palavras, com objetivo de correção e otimização do texto filtrado.
- `trim`¹⁸: retira espaços em branco desnecessários.
- `substr_count`¹⁹: é a principal função na mineração textual, é usada para contabilizar as comparações entre textos ou fragmentos de textos. Conta o número de ocorrências de alguma expressão no texto. Como essa contagem não tem controle, pode retornar um resultado não confiável caso o texto não tenha sido filtrado anteriormente (esta função, em conjunto com a estrutura do algoritmo, pode retornar os chamados falsos positivos).

A seguir, é visualizada na figura 6, a primeira função codificada que realiza um suporte inicial, para que se possa realizar na sequencia a mineração textual.

```
function filtroTextual($texto){
    $apagarPalavras = ' ';
    // abaixo segue um prototipo de dicionario que
    // retira as suas expressoes de um determinado texto:
    $dicionario = array('estudar','disponibilizar','fornecer',' cada ','outros','etc',
        ' que ',' de ','De',' em ','Em',' para ','Para','um','Um',
        ' desde ','Desde',' a ',' e ',' i ',' o ','-',' ',' ',' da ','
        ' uso ','através','empreguem','!','?');
    $formatado = str_replace($dicionario,$apagarPalavras,$texto);

    $textoFormatado = explode(' ', $formatado);

    $quantia = count($textoFormatado);
    $textoFiltrado = '';
    for($i=0;$i<$quantia;$i++){
        $tamanho = strlen($textoFormatado[$i]);

        if($tamanho > 2){
            $textoFiltrado .= $textoFormatado[$i].' ';
        }
    }
    $textoFiltrado = trim($textoFiltrado, " ");
    return $textoFiltrado;
}
```

Figura 6. Filtro textual em PHP.

Como pode ser visualizado na figura 6, foram também usados mecanismos da própria linguagem PHP, concebido um início de estrutura de dicionário, em que são guardadas os

15 http://www.php.net/manual/pt_BR/book.strings.php

16 http://www.php.net/manual/pt_BR/function.str-replace.php

17 http://www.php.net/manual/pt_BR/function.explode.php

18 http://www.php.net/manual/pt_BR/function.trim.php

19 http://www.php.net/manual/pt_BR/function.substr-count.php

termos que precisam ser ignorados para as combinações.

Para as combinações, todos os textos precisam receber um tratamento, já que as funções do PHP para *strings*, realizam combinações de palavra por palavra, pontuação a pontuação. Com isso, pode gerar falsos positivos e, o dicionário foi concebido para no mínimo evitar grandes erros de combinações e elevar a confiança no projeto.

Este pequeno dicionário serve como uma espécie de filtro aplicado em todos os textos ou palavras-chave, aumentando a confiança na resposta do sistema, entretanto ainda existem lacunas que podem ser melhorados com um estudo mais aprofundado sobre mineração textual e desenvolvimento de algoritmos de interpretações da linguagem humana.

A fórmula concebida para ΣH_p é definida como:

- Σ (total de combinações textuais positivas x P_u).

As competências profissionais destinam-se a pontuação por meio da experiência profissional. Esta pontuação também já ocorre nos processos organizacionais de seleção atualmente levando em conta experiência dentro e fora da área de formação. Para este protótipo foi considerado na totalidade sem discriminação de área de conhecimento.

Por meio da fórmula:

```
//formacao do score do nivel de experiencia profissional COMPETENCIAS_PROFISIOAIS
if(!empty($profissionalProfessor)){
    if($this->pesoProfissional != 0){

        $quantiaProfissoes = count($profissionalProfessor);
        for($w=0;$w<$quantiaProfissoes;$w++){
            $pontuacaoProfissao += $this->pesoProfissional +
                ($profissionalProfessor[$w]['experiencia'] * $this->valorPontuacaoPadrao);
        }
    }
}

//formacao de score da HABILIDADES_PESSOAIS
if(!empty($habilidadesProfessor)){
    if($this->pesoHabilidade != 0){

        $quantidadeMinerada = $this->mineracaoTextualHabilidades($disciplinaFiltrada,$habilidadesProfessor);
        $pontuacaoHabilidades += ($quantidadeMinerada * $this->pesoHabilidade);
    }
}
}
```

Figura 7. Codificação de fórmulas para competências profissionais e habilidades pessoais.

- $\Sigma C_p = \Sigma ((\text{anos de atuações profissionais} \times \text{valor padrão}) + P_u)$.

Na figura 7 é visualizado a codificação para as competências profissionais e as habilidades pessoais (além das equações, há o emprego da contabilização oriunda da mineração textual).

Os anos de experiência são calculados já na importação do currículo do professor, baseado no tempo de permanência nas atividades profissionais (somente considera as atividades profissionais após a sua data da primeira graduação). O valor padrão é uma pontuação básica existente no processo seletivo.

Para as competências didáticas foram definidos dois atributos: os níveis de formação e a mineração textual entre os cursos das quais os professores têm em suas formações e as descrições da disciplina alvo.

No primeiro atributo foi usado como referência o próprio processo seletivo discutido com os gestores formalizado como $Cd = \sum Ng + \sum Mtd$. Ng os níveis de graduação e Mtd é a mineração textual com a disciplina. Desta forma, a equação é disposta da seguinte forma:

- $\sum Ng = \sum (\text{pontuação normativa para o nível de graduação} + Pu)$;
- $\sum Mtd = \sum (\text{total de combinações textuais positivas})$.

A mineração textual ocorre de forma parecida na quantificação de habilidades pessoais descritos anteriormente. A diferença é que o mecanismo realiza um confronto entre as descrições existentes no título e ementa da disciplina com o título do professor (o nome do curso da formação, desde a graduação até pós-doutorado).

```
//formacao de score do nível de formacao - GRADUACAO
$quantiaGraduacao = count($graduacaoProfessor);
for($j=0;$j<$quantiaGraduacao;$j++){

    if($this->pesoGraduacao != 0){

        if($graduacaoProfessor[$j]['tipo_graduacao'] == 'graduacao'){
            $pontuacaoGraduacao += ($this->pesoGraduacao + $graduacaoRegulamento);
        }elseif($graduacaoProfessor[$j]['tipo_graduacao'] == 'especializacao'){
            $pontuacaoGraduacao += ($this->pesoGraduacao + $especializacaoRegulamento);
        }elseif($graduacaoProfessor[$j]['tipo_graduacao'] == 'mestrado'){
            $pontuacaoGraduacao += ($this->pesoGraduacao + $mestradoRegulamento);
        }elseif($graduacaoProfessor[$j]['tipo_graduacao'] == 'doutorado'){
            $pontuacaoGraduacao += ($this->pesoGraduacao + $doutoradoRegulamento);
        }else{
            $pontuacaoGraduacao += ($this->pesoGraduacao + $posdocRegulamento);
        }
    }

}

$quantidadeMinerada = $this->mineracaoTextualDisciplina($graduacaoProfessor[$j]['curso'],$disciplinaFiltrada);
$pontuacaoGraduacao += $quantidadeMinerada;
```

Figura 8. Codificação de fórmulas para os níveis de formação.

Nesta figura 8 é visualizada toda a codificação das equações empregadas nos cálculos de recomendação, em conjunto com a contabilização oriunda da mineração textual. É aceito uma pontuação somente de acordo com o nível de formação que o candidato a professor possui.

Após todos os cálculos, é finalizada a equação global ($R_p = \sum Cd + \sum Cp + \sum Hp$) para um professor específico e assim é realizado para todos os professores existentes na base de dados ou previamente cadastrados para a disciplina alvo. O escopo de uma forma geral não é fechado e o sistema é condicionado ao título de “beta eterno”, pois os conjuntos aqui interpretados podem ser melhorados e inclusive criadas outras equações para outras informações, qualitativas ou quantitativas, a fim de melhorar continuamente e adaptar-se à gestão organizacional.

Essa denominação equivale também ao planejamento estratégico da instituição (seção 2), pois precisa constantemente estar em adaptação conforme a ambientação onde se insira o *software*. Para a ação planejadora do gestor esta ferramenta fornecerá o auxílio ao planejamento de alocação de professores e o protótipo já disponibiliza mecanismos que ajudem a acelerar a tomada de decisão do gestor.

9. Conclusões

Por meio dos estudos realizados para este trabalho verificou-se o nível de capacidade que os

sistemas informacionais possam alcançar, sejam estes voltados a atividades em setores humanos ou em exatas. Os conceitos de gestão podem fornecer um roteiro de ação para que o desenvolvedor de *software* saiba conceber sistemas mais especialistas ao objetivo fim da aplicação. Não somente desenvolver sistemas que automatizam atividades diárias, contudo, podendo ser úteis também para a tomada de decisão humana.

Nesta decisão humana implica não na pretensão do *software* resolver problemas automaticamente para as pessoas já que as soluções nem sempre são lineares ou exatas. Mas sim de modelos computacionais que visam atender modelos organizacionais e constantemente adaptar-se as mudanças temporais sem perder sua capacidade de auxílio de gerenciamento ou de valores em suas respostas.

O sistema de recomendação é focado neste auxílio, baseado no que o indivíduo precisa, consegue fornecer sugestões para dar origem a uma decisão pelo usuário. A gestão do conhecimento não deve ser encarada como obstáculo. Deve ser visto como aliada na contextualização entre *software* e pessoa, entre sistema e estratégia ou, ainda, entre gestor e ferramenta. De um lado uma consciência e, de outro, um sistema que possa acompanhar e valorizar o capital humano.

A inteligência artificial mostrou-se ser um universo à parte para a concepção de sistemas cada vez menos manipuláveis pelo usuário, mas retornando cada vez mais informação especializada para o próprio usuário. Isso pode ser observado no painel de controle da inferência e até onde determinam-se as pontuações e minerações de informações para a sugestão. Outros estudos, mais focados e aprofundados podem ser gerados a partir deste projeto (seção de Trabalhos Futuros). Em que se possa conceber uma plataforma geradora de conhecimento para a instituição acadêmica. Desde já os conceitos básicos mostraram-se fazer parte do dia a dia na tomada de decisão dos diretores de área e precisam ser inerentes aos processos de codificação de *software* mais especialistas.

Ainda há muito que estudar sobre a ontologia. Entretanto, já disponibiliza mecanismos novos para o gerenciamento da compreensão e definição do conhecimento. Estes mecanismos podem ser úteis na gestão organizacional no objetivo de separar aquilo que faz parte da cultura do indivíduo, para o conceito que verdadeiramente faz parte da gestão acadêmica. Também mostrou-se como pode ser útil no desenvolvimento de aplicação voltada para o conhecimento especializado, e a comunicação, mesmo que em vias de *software*, entre todos os indivíduos, não importando se estes estão em culturas, setores, operações ou profissões diferentes.

Assim, houve uma compreensão maior sobre o desenvolvimento de *software* e até onde os Sistemas de Informação podem chegar, ultrapassando até mesmo outros *softwares* menos especialistas em determinadas ações estratégicas. É importante ressaltar que os sistemas de informação também nunca estarão concluídos, sempre haverá a necessidade de adaptar-se as culturas organizacionais, às infraestruturas, e com elas continuamente manter-se o seu desenvolvimento e atualização, seja em codificação ou em conceito. Aumenta também a confiança que um gestor possa ter no *software* viabilizando uma tomada de decisão segura e em conjunta com modelos algoritmos.

10. Referências

Barcellos, Carla Duarte. Musa, Daniela Leal. Brandão, André Luiz. Warpechowski, Mariusa. Sistema de Recomendação Acadêmico para Apoio a Aprendizagem. (2007) <http://www.cinted.ufrgs.br/ciclo10/artigos/3fDaniela.pdf>, Setembro.

Bittencourt, Guilherme. Inteligência Artificial: ferramentas e teorias. Florianópolis : 2. ed.

UFSC, 2001.

- Burke, Robin. Hybrid Recommender Systems: Survey and Experiments. Department of Information Systems and Decision Sciences. California State University. Fullerton, California, United States. Kluwer Academic Publishers. (2002) <http://portal.acm.org/citation.cfm?id=586352>, Setembro.
- Carneiro, Raquel Elias. Brito, Parcilene Fernandes de. Definição de uma Ontologia em OWL para Representação de Conteúdos Educacionais. VII Encontro de Estudantes de Informática do Estado do Tocantins, 2005. Anais... (2005) http://www.arquivar.com.br/espaco_profissional/sala_leitura/teses-dissertacoes-e-monografias/Definicao_de_uma_Ontologia_em_OWL_para_Representacao_de_Conteudos_Educacionais.pdf , Novembro.
- Coelho, Márcio. A essência da administração: conceitos introdutórios. São Paulo: Saraiva, 2008.
- Dutra, Joel Souza. Gestão de pessoas : modelo, processos, tendências e perspectivas. São Paulo: Atlas, 2002.
- Gonçalves, Julia Aparecida. Souza, Renato Rocha. Relações e conceitos em ontologias: Teorias de Farradane e Dahlberg. Seminário de Pesquisa em Ontologia no Brasil. Universidade Federal Fluminense. Departamento de Ciência da Informação. Rio de Janeiro. (2008) <http://www.uff.br/ontologia/artigos/15.pdf>, Novembro.
- Jané, Dario de Almeida. Uma Introdução ao Estudo da Lógica Fuzzy. Revista de Humanidades e Ciências Sociais Aplicadas, Ourinhos/SP, No 02. (2004) http://www.faesio.edu.br/horus/artigos%20anteriores/2004/artigo_dario.pdf, Novembro.
- Lichtnow, Daniel. Loh, Stanley. Kampff, Adriana Justin Cerveira. Primo, Tiago. Garin, Rodrigo Saldaña. Oliveira, José Palazzo Moreira de. Lima, José Valdeni de. Palazzo, Luiz A. Moro. O Uso de Técnicas de Recomendação em um Sistema para Apoio à Aprendizagem Colaborativa. (2006) Disponível em <http://ww.inf.ufrgs.br/adapt/artigos/06%20RBIE.pdf>, Setembro.
- Luger, George F. Tradução de Engel, Paulo Martins. Inteligência Artificial: Estruturas e estratégias para a solução de problemas complexos. 4. ed. São Paulo : Artmed, 2002.
- Migliori, Magda Bianchini. Recursos Humanos e Gestão do Conhecimento. 2006. Tese (Doutorado em Ciências Empresariales) - Pós-Graduação em Administração. Universidad del Museo Social Argentino, Chapecó, 2006.
- PHP. Manual da linguagem de programação PHP. (2010) <http://php.net>, Julho.
- Rich, Elaine. Tradução de Newton Vasconcellos. Inteligência Artificial. São Paulo : McGraw-Hill, 1988.
- Shaw, Ian S. Tradução de Simões, Marcelo Godoy. Controle e modelagem Fuzzy. São Paulo: Edgar Blucher, 1999.
- Tachizawa, Takeshy; Andrade, Rui Otávio Bernardes de. Gestão de Instituições de Ensino. 1a. ed. São Paulo : Fundação Getúlio Vargas, 1999.