

UNIVERSIDADE PRESBITERIANA MACKENZIE

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARCIO PORTO FEITOSA

Um Modelo Computacional de Apoio à Tomada de
Decisão em Processos de Avaliação de Pessoas por
Competências

São Paulo

2009

MARCIO PORTO FEITOSA

**Um Modelo Computacional de Apoio à Tomada de
Decisão em Processos de Avaliação de Pessoas por
Competências**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Presbiteriana Mackenzie, como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica na área de Engenharia da Computação.

Orientador: Prof. Dr. Nizam Omar

São Paulo

2009

Em memória ao meu pai que sempre pregou que o estudo abre mais portas que o aríete.

*Um desempenho superior depende de um
aprendizado superior (Peter Senge)*

AGRADECIMENTOS

A Deus, por nos ter concedido o privilégio da existência e a vontade de decifrá-la.

Ao Instituto e à Universidade Presbiteriana Mackenzie por terem contribuído de forma significativa na minha formação acadêmica nas diversas etapas da minha vida.

À minha esposa Yara, verdadeira responsável pela realização deste trabalho.

Aos meus filhos, Leonardo e Virgínia, que compreenderam a minha ausência em momentos importantes devido à minha dedicação aos estudos.

À minha mãe e irmã pelos constantes apoio e carinho.

Ao professor Nizam Omar que me conduziu com paciência e sabedoria e me mostrou o caminho.

Às professoras Sandra e Pollyana que me fizeram ver com mais sensibilidade o lado humano por trás da tecnologia.

Ao professor Maurício Marengoni pelos excelentes conselhos e ensinamentos.

Ao professor Luiz Henrique por ter o dom (e a humildade) de descer do alto do seu conhecimento - muito distante para nós alunos – e nos fazer entender, de fato, sua complexa matéria.

Aos demais professores pelo profissionalismo, consideração e carinho com que nos conduziram a este novo patamar do conhecimento.

À CAPES pelo apoio financeiro e principalmente por acreditar que alguém mais próximo da aposentadoria do que do início de carreira ainda tem muito a contribuir com o progresso da ciência.

A todos os meus colegas de classe.

A todas as demais pessoas que colaboraram direta ou indiretamente neste trabalho.

RESUMO

Este estudo tem como tema a idealização de um modelo computacional que represente as pessoas pelas suas competências profissionais e comportamentais, de forma a sugerir um padrão de medida para que profissionais academicamente formados, profissionais em formação (alunos), empresas e instituições de ensino tenham um mesmo referencial para aferição de conhecimentos, habilidades e atitudes de determinada pessoa (ou grupos de pessoas) . O estudo está embasado na teoria da “Gestão por Competências” que se apresenta em franca utilização e expansão nas áreas de Recursos Humanos das empresa. O estudo objetiva, não só a idealização teórica, mas a implementação prática de protótipo de um Portal Web com as funcionalidades básicas de cadastro e busca de informações de interesse, tanto por pessoas como por oportunidades oferecidas. A metodologia da pesquisa consistirá, após pesquisa teórica do tema, idealizar um modelo computacional para a Competência, a implementação de uma base de dados que agregue as funcionalidades básicas necessárias ao Portal Web, a confecção das páginas Web para funcionamento do protótipo e testes de funcionalidade e desempenho, sendo que os resultados esperados se situam em três setores: adequação do modelo, funcionalidade do Portal e desempenho computacional.

Palavras chave: Competência; Gestão por Competências; Avaliação.

ABSTRACT

The main theme of this work is the construction of a computational model that represents the people for their professional and behavioral skills, suggesting a gauge in order to enable academically trained professionals, students, companies and schools to have the same benchmark for measuring knowledge, skills and attitudes of a certain person (or groups of people). The study is based on the theory of "Management by Competencies", that is nowadays being much utilized and its use is increasing within the Human Resources area of the companies. The study aims not only the theoretical idealization but also the practical use of a prototype of a Web Portal with the basic features of register and search for information of interest, by people as well as by opportunities that are offered. The methodology utilized in the research, after theoretically researching the subject, will be to construct a computational model for Competency, the implementation of a database that can gather the basic functions required by the Web Portal, the construction of the Web pages for the prototype functioning and functionality and performance tests. The expected results are in 3 areas: adequacy of model, Portal functionality and computational performance.

Key words - Competency, Competence-based Management, Evaluation.

SUMÁRIO

1- INTRODUÇÃO	1
1.1- OBJETIVOS DA PESQUISA.....	2
1.2- APRESENTAÇÃO DOS DE MAIS CAPÍTULOS.....	3
2- GESTÃO POR COMPETÊNCIAS	4
2.1- GESTÃO POR COMPETÊNCIAS NO BRASIL.....	6
2.2- O PROBLEMA.....	7
3- MODELO CONCEITUAL DE COMPETÊNCIA E CONSIDERAÇÕES PRÁTICAS	11
3.1- AVALIAÇÃO POR COMPETÊNCIAS.....	12
3.2- PADRÕES ESTRUTURAIS PARA MODELAGEM DE COMPETÊNCIAS.....	17
3.3- O ARMAZENAMENTO DE VALORES NO MODELO PROPOSTO DE SKILL.....	22
4- MODELO COMPUTACIONAL PROPOSTO E TECNOLOGIAS NECESSÁRIAS	25
4.1- PLATAFORMA TECNOLÓGICA E AMBIENTE DE DESENVOLVIMENTO.....	26
4.2- A BASE DE DADOS.....	26
4.2.1- O Modelo do SKILL	30
4.2.2- O Modelo da Pessoa e respectivos SKILLS	32
4.2.3- O Modelo da Oportunidade e Respective SKILLS	34
4.2.4- Tabelas de Referência	36
4.3- LINGUAGEM SQL APLICADA AO MODELO.....	36
4.3.1- A busca por múltiplos SKILLS – uma Competência	37
4.4- CASOS DE USO.....	40
4.5- DIAGRAMA DE CLASSES E A CRIAÇÃO DOS OBJETOS.....	41
4.6- A QUESTÃO DA EXPLOSÃO DE CLASSES E A SÉRIE 9000.....	45
4.7- OS CASOS DE USO DE “CADASTRAMENTO”.....	47
4.8- OS CASOS DE USO DE “BUSCA”.....	51
5- CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	60
REFERÊNCIAS	62
APÊNDICE A - Pesquisa preliminar na Internet e via e-mail para verificação aproximada da situação atual da utilização de Bancos de Competências no Brasil	64
APÊNDICE B - Projeto e Normalização do Modelo de Dados Relacional - Pessoas e SKILLS.....	67
ANEXO A - Padrão HR-XML – Exemplos.....	74

1 INTRODUÇÃO

O rápido desenvolvimento tecnológico e as mudanças crescentes nos perfis dos profissionais necessários para atuarem em qualquer área profissional, em especial a área da Tecnologia da Informação, têm levado, por um lado as pessoas a buscarem cada dia mais novas capacitações, e por outro as instituições de ensino têm buscado aumentar e diversificar os tipos de cursos e as estratégias didáticas para a constante manutenção dos níveis de competência desses profissionais.

No meio empresarial, atualmente, o patrimônio representado pelos conhecimentos dos colaboradores diretos ou funcionários adquiriu mais significância, de forma que a Gestão de Pessoas torna-se a cada dia fundamental para o sucesso de qualquer empreendimento empresarial.

Como se encontram diversas opções ao se escolher, por exemplo, uma Formação, aqui entendida como a realização de um curso, seja ele formal ou não, essa decisão precede-se de uma atividade de avaliação para ajudar na melhor escolha. Por outro lado, o empregador, ao se decidir por um colaborador, necessita determinar qual é a melhor pessoa em um grupo para ocupar uma determinada posição.

As escolhas de profissionais, o oferecimento de formações e a capacitação individual, apesar de, em conjunto objetivarem o mesmo fim, não possuem meios e mecanismos comuns de inter-relacionamento e de cooperação. A proposta de um Modelo Computacional que opere sob a forma de um Portal Web de apoio a processos de Gestão por Competências, atendendo a três entidades - pessoas, empresas e escolas - abrangeria todo o domínio envolvido.

Avaliar pessoas é um processo complexo pela diversidade de fatores, alguns no domínio discreto, outros num domínio contínuo de valores, resultando em uma quantidade de perfis pessoais semelhantes, mas não iguais, dificultando bastante a comparação entre eles.

Avaliações errôneas levam projetos a resultados indesejados, seja atrasando-os em seus cronogramas de execução, seja resultando em decréscimo de qualidade ou perda de funcionalidades. A abrangência do universo do estudo aumenta ainda mais a complexidade do problema. Um profissional pode ser considerada excelente em uma pequena empresa, mas não atender aos requisitos básicos exigidos por empresas maiores. Ou um aluno aprovado com méritos em uma escola com

deficiências de ensino não teria a mesma avaliação em uma escola de estrutura mais rigorosa.

Segundo levantamentos preliminares na Internet, através de ferramentas de busca, e consulta via e-mail aos departamentos de RH de 11 empresas de grande e médio portes na cidade de São Paulo (vide APÊNDICE A), a despeito dos padrões especificados atualmente (IMS, 2002; HR-XML, 2007; IEEE, 2005), não foi identificado modelo computacional implementado, ou mesmo teoricamente idealizado, que resolva este problema e que também proponha padrões que sejam comuns às três entidades em questão.

1.1 OBJETIVOS DA PESQUISA

O objetivo geral deste estudo é criar um modelo conceitual e respectivo modelo computacional de um ambiente de apoio à tomada de decisão na atividade de avaliação de pessoas por Competências. Neste cenário estão envolvidas três entidades: Empresas, Escolas e Pessoas que interagem através de uma entidade abstrata intitulada Profissional. O Profissional é a Pessoa com os requisitos necessários a desempenhar determinada função que uma determinada Empresa necessita, sendo também o “título” que a Escola pretende que suas Pessoas (alunos) adquiram para terem condições de concorrer por uma das oportunidades oferecidas pelas Empresas no mercado de trabalho. A Pessoa em si é o aspirante a Profissional, ou a Profissional mais qualificado, procurando cursos profissionalizantes de formação e de especialização, como também para certificações, e também se esforçando na obtenção de experiências profissionais mais relevantes.

Para que o modelo atinja a abrangência pretendida, deverá ter a forma de um portal Web com as seguintes funcionalidades básicas:

Para a empresa:

- Publicar uma oportunidade de trabalho com os respectivos requisitos;
- Localizar pessoas cujas competências se enquadrem em determinada oportunidade publicada;

Para o profissional:

- Cadastrar-se no portal e relacionar seus atributos e competências.

- Localizar oportunidades que estejam próximas do seu perfil de competências;
- Verificar que competências lhe faltam para que possa concorrer a determinada posição no mercado.

Para a escola:

- Acompanhar as competências solicitadas pelo mercado para verificar o alinhamento com seus cursos oferecidos.
- Comparar as competências solicitadas pelo mercado com as dos seus alunos.
- Comparar as competências dos seus alunos com as de outras escolas em cursos equivalentes.

Dada a abrangência do domínio, o foco deste estudo concentrar-se-á no conjunto Profissional-Empresa, ficando o sub-domínio acadêmico (Escola) sugerido em “Trabalhos Futuros”, sem perder, no entanto, a coerência na modelagem de forma geral para que o acoplamento futuro se viabilize sem problemas impactantes.

1.2 APRESENTAÇÃO DOS DEMAIS CAPÍTULOS

Este capítulo teve o objetivo de apresentar o problema, que será melhor detalhado adiante, indicar os objetivos da pesquisa a ser desenvolvida e as motivações que levaram a ela. No Capítulo 2 será apresentada a Gestão por Competências e como hoje ela é vista no meio empresarial. No capítulo 3 será estudado o modelo conceitual da Competência e os principais padrões de projeto para o desenvolvimento de soluções baseadas em competências. No capítulo 4 será apresentado o modelo computacional, objetivo deste estudo, e as tecnologias necessárias para implementação. No capítulo 5 será apresentado o fluxo operacional do protótipo implementado associado aos conceitos estudados e os fatores determinantes do *ranking* na lista dos candidatos possíveis, o *Skill Score* e o *Skill Gap*. Os capítulos 6 e 7 apresentam as Conclusões e as propostas para Trabalhos Futuros.

2 GESTÃO POR COMPETÊNCIAS

Gestão por Competências é uma tendência do mercado a partir do início dos anos 90 (DUTRA et al, 2001) (RABAGLIO, 2001). As empresas atualmente trabalham no sentido de avaliar seus talentos internos e também de buscar profissionais no mercado com base neste modelo porque se conseguiu dar maior precisão aos processos de avaliação de pessoas sem comprometer a essência dos processos inerentes aos setores voltados à Gestão de Recursos Humanos.

Para que se possa entender o significado de Competência, serão apresentadas algumas definições geralmente aceitas para o termo. Perrenoud (2000) ao estudar a relação que existe entre Formação e Competências, coloca explicitamente que a formação continuada está a serviço do desenvolvimento das competências profissionais, pois algumas formas de aperfeiçoamento ampliam a capacidade de profissionais de realizarem suas tarefas e com isso “esperar que isso desenvolva também suas competências profissionais”. Cabe ao interessado incluir essas características em suas habilidades e atitudes. Para ele “Uma competência é um saber-mobilizar, trata-se (...) de uma capacidade de mobilizar um conjunto de recursos (...) a fim de enfrentar com eficácia situações complexas e inéditas”.

Qualquer conhecimento continuará virtual até que ele seja absorvido ou mobilizado a favor da sua capacidade de utilizá-lo adequadamente. O mesmo ocorre no processo de avaliação formativa, não se pode afirmar que qualquer formação continuada leva por si só à construção de competências.

Soluções para a gerência de competências surgem dos mais diversos ramos de atividades: instituições de ensino a utilizam com o foco de educar e formar pessoas; empresas, como instrumentos para a gestão de RH; e fornecedores de treinamento corporativo, que enxergam competências como requisitos e resultados do uso de um conteúdo instrucional.

Encontram-se inúmeras definições para o termo “Competência”. Clark (2005), em sua tese de doutorado, cita 18 definições, mas a definição atualmente adotada no segmento corporativo de Recursos Humanos foi apresentada por Scotty B. Parry em 1996, resultado da síntese de diversas sugestões que foram apresentadas na conferência sobre competências em Johannesburg (África do Sul) em 1995 por especialistas em Recursos Humanos.

Nas palavras dele (PARRY, 1996):

“A competency is: A cluster of related knowledge, skills and attitudes (K, S, A) that affects a major part of one's job (a role or responsibility), that correlates with performance on the job, that can be measured against well-accepted standards, and that can be improved via training and development.”

E na precisa tradução de RABAGLIO (2004, pag. 12)

“agrupamento de conhecimentos, habilidades e atitudes correlacionadas, que afeta parte considerável da atividade de alguém, que se relaciona com o seu desempenho, que pode ser medido segundo padrões pré-estabelecidos, e que pode ser melhorado por meio de treinamento e desenvolvimento.”

No entanto, pela natureza complexa dos elementos envolvidos, a definição não é capaz de abranger todo o domínio. Segundo Dutra (2001, pag 28) “A competência é compreendida por muitas pessoas e por alguns teóricos da administração como um conjunto de conhecimentos, habilidades e atitudes necessários para que a pessoa desenvolva suas atribuições e responsabilidades.”, no entanto, esta definição, não garante que o detentor dos atributos de uma dada Competência aja com competência em determinada situação. É o que também cita Rabaglio (2001), com relação à diferença entre “ter competência” e “ser competente”.

Para efeito de modelagem computacional, a definição de Competência de Scotty B. Parry, apresentada acima, devido à sua idealização baseada na aplicação prática empresarial, tem uma estrutura mais facilmente associável à modelagem em si, sendo adotada como definição-base neste estudo.

Padrões para a modelagem computacional de competências são encontrados hoje em fóruns de discussão e padronização e buscam estabelecer, dentro de comunidades, bases comuns para o desenvolvimento de sistemas baseados em competências que possam interagir entre si. Alguns desses padrões são os propostos pelo IMS [2002], pelo IEEE [2005] e pelo HR-XML [2007]. O primeiro tem seu foco nas atividades educacionais, porém é genérico e extensível podendo ser usado como modelo inicial para a aplicação em outros enfoques. O segundo tem seu foco em conteúdos (objetos de aprendizagem). E o terceiro, para a gestão de recursos humanos. Pelo seu enfoque mais prático na avaliação de pessoas, o padrão HR-XML será tomado como base principal na modelagem do sistema proposto.

2.1 GESTÃO POR COMPETÊNCIAS NO BRASIL

Segundo Fleury (2000), a Gestão por Competências foi inicialmente adotada nas empresas brasileiras como uma mera mudança de rótulo. A definição de Scotty B. Parry pela facilidade de operacionalização foi simplesmente adaptada aos procedimentos correntes, em muitos casos implicando em simples mudanças “burocráticas” nos procedimentos para avaliação de indivíduos.

Também foi percebido durante o estudo, que muitos dos tradicionais Bancos de Currículos tiveram seus títulos alterados para “Banco de Competências” sem que houvesse significativa modificação de contexto (constatação embasada também na experiência do autor pela vivência em consultoria de recursos humanos e desenvolvimento de sistemas).

Em levantamento preliminar pela Internet e consulta via e-mail a alguns gestores de departamentos de recursos humanos de 11 empresas de grande e médio portes na cidade de São Paulo, foi percebido que, aparentemente, há muitos casos de mera mudança de “rótulo”, sendo que a estrutura ainda é a antiga/tradicional. Apenas um único caso, uma entidade acadêmica, aparentou ter uma base de dados com as competências dos seus alunos para acesso das empresas interessadas em contratar estagiários. Quanto à consulta via e-mail, apesar de a grande maioria considerar o tema de relevada importância, aproximadamente 1/3 ainda não adota Gestão por Competências, 1/3 está em processo de implantação e os restantes 1/3 declararam já ter implantado. Quanto às respostas acerca dos sistemas de bases de dados modelados por Competências, apenas uma única empresa declarou ter este recurso, mas não houve a possibilidade de verificação *in-loco*.

Como os recursos informatizados acabam surgindo à medida das necessidades de apoio aos fluxos de informação das empresas, conclui-se que pelo estado em que estão as implantações da linha administrativa da Gestão por Competências, segundo apurado na entrevista via e-mail, o surgimento de bases de dados modelados para tal ainda estão por vir, de forma que este estudo certamente virá a contribuir de forma significativa na concepção desses sistemas que mais cedo ou mais tarde se farão necessários dentro das organizações.

2.2 O PROBLEMA

As empresas, ao contratarem novas pessoas para trabalharem em seus quadros de colaboradores, procedem a processos seletivos cujas etapas são predominantemente avaliativas (TSUNO, 2007). Há, no entanto, diferentes enfoques nesses processos, sendo os mais usuais (1) a busca pelo melhor dentro de um determinado grupo e (2) a busca pelo mais adequado à determinada tarefa. O primeiro caso foca-se em determinar as pessoas mais capazes de realizar mais tarefas ou realizar tarefas da melhor forma. Os parâmetros usualmente utilizados são dinamismo, inteligência, pró-atividade, etc. O segundo caso foca-se na tarefa e busca alguém que se adapte bem a ela e gere resultados eficazes. Uma função, por exemplo, que requeira concentração na análise de dados, em grande parte do dia, para produzir relatórios consolidados, possivelmente seja melhor realizada por uma pessoa mais focada à mesa de trabalho o que não ocorreria com uma dinâmica e adepta a relacionamentos.

Toda decisão pressupõe a existência de mais de uma possibilidade de escolha. No mínimo decide-se fazer ou não fazer alguma coisa. Por se ter mais de uma possibilidade de ação, via de regra, procede-se a uma atividade de avaliação para ajudar na melhor escolha. Avaliar significa, segundo os principais dicionários da língua portuguesa, “estabelecer a valia, o valor ou o preço de” ou também “determinar a quantidade de; computar”, o que significa que com a atividade de avaliação vai-se chegar a valores, sejam de maior ou menor precisão, que darão aos decisores um apoio para a escolha.

Avaliar pessoas, no entanto, envolve mais complexidades do que avaliar, por exemplo, um determinado tipo de máquina. Pelo universo finito de fatores, na avaliação de máquinas é possível a elaboração de planilhas comparativas, o que já não é tão simples com as pessoas. A combinação de fatores é maior e a tabulação torna-se complexa, ou, em um esforço de simplificação, torna-se imprecisa, dificultando a comparação entre os indivíduos avaliados.

Outro aspecto do problema seria a “rotulação” incompatível com o “produto” (objeto ou pessoa), induzindo o interessado a adquirir algo que não estaria apto a colaborar na solução do seu problema, ou, no lado oposto, a não se interessar por algo de grande valia. Em todas as situações sempre há algum tipo de prejuízo. Seja

para o comprador, seja para o produto. Seja para o contratante, seja para o contratado.

Há ainda a questão da abrangência do domínio do estudo, de forma que o nosso futuro modelo computacional, para incluir determinada pessoa na relação dos candidatos “competentes”, terá que se apoiar em parâmetros relativos ao contexto.

A figura 1 a seguir retrata o cenário do processo de gestão de pessoas que hoje existe e que não dispõe de um mecanismo integrado que ofereça suporte ao processo como um todo.

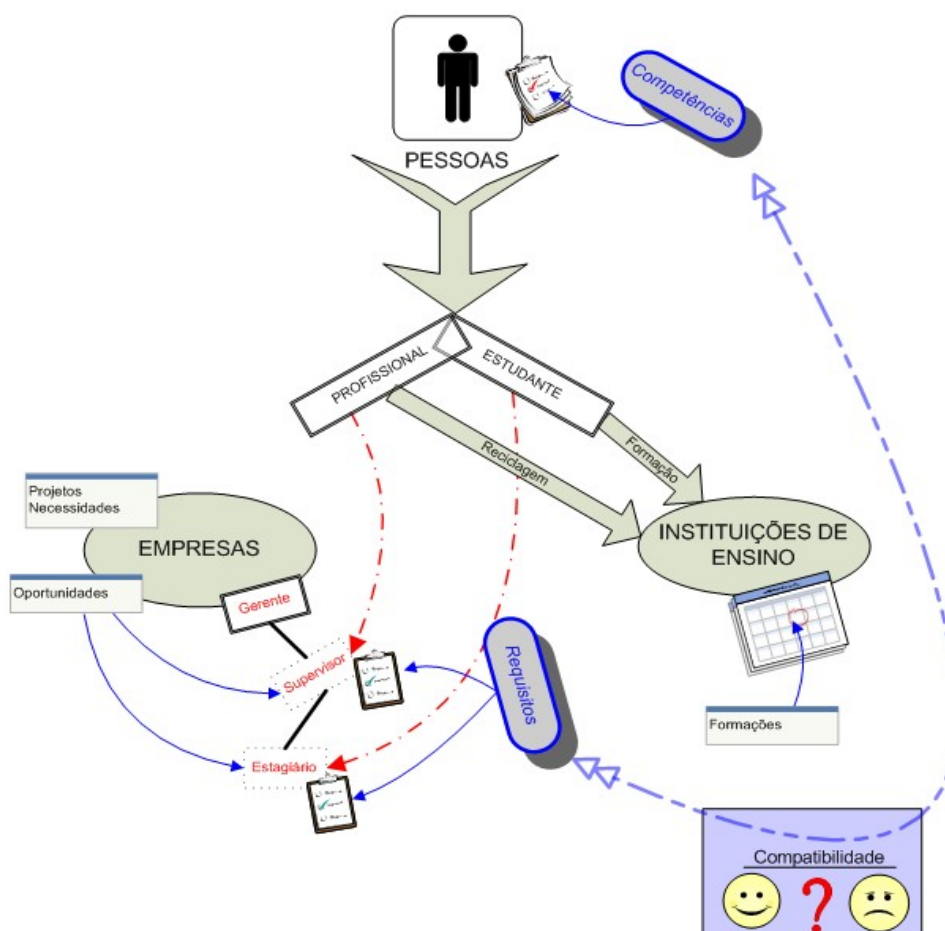


Figura 1 – Cenário do Processo de Gestão de Pessoas

O processo representado na figura 1 mostra uma série de problemas que necessitam ser abordados. As três entidades, Pessoas, Instituições de Ensino e Empresas, relacionam-se através das entidades representadas pelo “Profissional”, pessoa com determinado grau de formação e apta a desempenhar uma tarefa requerida pelo mercado de trabalho, e pelo “Aluno”, pessoa em formação acadêmica, seja em nível técnico, superior ou até mesmo em cursos de formação

especialmente elaborados para projetos de inclusão social. As empresas tem seus Projetos e Necessidades e para levá-los a cabo precisam encontrar profissionais aptos ou com as devidas competências que são expressadas por meio de Requisitos às vagas. As competências estão nas pessoas, que se estiverem alinhadas aos requisitos empresariais, serão enquadradas na categoria de “profissional apto” ou “profissional competente” (para os referidos requisitos). As Instituições de Ensino visam formar profissionais com as competências necessárias para suprir as necessidades do mercado de trabalho (e também trazer novas idéias e melhorias), sejam alunos recém-formados, sejam profissionais em cursos de reciclagem.

O custo de uma avaliação errada pode se revelar elevado. Um profissional recém colocado em uma posição demanda tempo para se adaptar a esta. Neste período o seu retorno para a empresa é menor que seu o custo, pois além de seu salário há os treinamentos, o tempo despendido pela chefia e por parceiros que o orientam durante este período, além do próprio tempo perdido pela empresa entre seu início e uma possível constatação de sua inadequação à posição, devendo-se reiniciar todo o processo de recolocação com outra pessoa. É sem dúvida nenhuma interessante que este profissional se mantenha na empresa e na posição designada *a priori* por um período considerado “mínimo”, e esta manutenção depende da asserção e o correto alinhamento entre o profissional e a posição ocupada.

A contratação de profissionais inadequados é uma constante nas empresas (TSUNO, 2007). Há ausência de parâmetros sólidos e padronizados. As entrevistas e testes são preparados sem critérios rigorosos e muitas conclusões são influenciadas pelo *feeling* do entrevistador, sem qualquer de base científica. A avaliação, então, pode acabar embasada em um conjunto de afirmações e opiniões que só poderão ser constatadas na experimentação prática, após a contratação do referido profissional. Só então será verificado se a escolha foi adequada.

Nas instituições acadêmicas, onde são encontrados os profissionais em formação que ocuparão as posições de estagiários nas empresas, assim como são encontrados os profissionais já formados em cursos de reciclagem, tais como os de pós-graduação, a questão da avaliação também tem problemas. Aqui a avaliação visa verificar se o aluno adquiriu o conhecimento necessário e está apto a evoluir para a próxima etapa. As salas de aula, em certos cursos, tem número grande de alunos, dificultado o acompanhamento do professor de forma individualizada, e o

nível de aquisição do conhecimento nem sempre é avaliado de forma satisfatória (LOCATELLI, 2007). As tradicionais notas, ou conceitos, praticados nas escolas, também não refletem com precisão o conhecimento do aluno, pois a nota pertence à disciplina e não aos itens específicos de conhecimento. Se tomarmos a nota de uma disciplina da área da computação que abranja modelagem de bases de dados e SQL não elucidará em que aspecto o aluno se saiu melhor ou pior. Uma empresa que necessite de um estagiário com bons conhecimentos em SQL não terá como saber se um determinado aluno realmente tem este conhecimento com base nesta nota referente a um conjunto mais amplo. Da mesma forma haverá a situação de contratar primeiro para se constatar os resultados depois. E na própria instituição acadêmica também haverá falta de consistência na definição dos pré-requisitos para o aluno cursar matérias posteriores, pois a nota da matéria como um todo não especifica que conhecimentos foram melhor adquiridos. Uma das possíveis soluções para este problema está no conceito do NAC, Nível da Aquisição do Conhecimento, em que os itens de conhecimento são associados a conceitos (PIMENTEL e OMAR, 2003), cujo protótipo foi implementado sob o nome de Sistema ACAvA – Ambiente Computacional de Avaliação da Aprendizagem (LOCATELLI, 2007) em que o modelo proposto por este estudo pretenderá dar suporte à integração ao modelo de Competências por intermédio do NAC.

3 MODELO CONCEITUAL DE COMPETÊNCIA E CONSIDERAÇÕES PRÁTICAS

Partindo-se do pressuposto que o ser humano possui uma estrutura comportamental complexa, que age e julga de forma variável de acordo com sua formação, suas certezas, suas inseguranças e também seu estado de espírito em determinado momento, pode-se concluir que a variação de parâmetros externos pode alterar significativamente o que uma determinada pessoa irá fazer ou decidir. Adicionando-se uma segunda pessoa para avaliar esta primeira tornará o problema complexo em termos de previsão de resultados confiáveis. Este é um dos problemas enfrentados pelos departamentos de Recursos Humanos das empresas de uma forma geral: avaliar de forma justa seus colaboradores atuais e avaliar corretamente os novos candidatos (TSUNO, 2007).

A avaliação de pessoas (por pessoas) começou a ficar um pouco mais precisa na década de 1950 quando surgiu a “Avaliação por Objetivos” (RABAGLIO, 2004). Este modelo, atribuído a Ralph W. Tyler, desenvolvido inicialmente para avaliar alunos nas escolas, foi adaptado para avaliar os profissionais nas empresas na época da linha administrativa intitulada “Administração por Objetivos”. Neste modelo, avaliador e avaliado negociavam previamente objetivos a alcançar em determinado espaço de tempo e, cumprido este, se reuniam para verificar o que foi e o que não foi cumprido. Desta forma conseguiu-se um parâmetro transformável em número, muito embora este número pudesse não representar com muita precisão a real competência da pessoa avaliada, uma vez que, se o estabelecimento inicial dos objetivos fosse condescendente, alguém menos competente poderia acabar sendo considerado mais competente que outra pessoa, por exemplo, que fosse mais competente que esta, mas que teve uma definição de objetivos bem mais rigorosa.

Outro modelo bastante utilizado na atualidade é a “Avaliação 360 graus”. Método em que a pessoa é avaliada não só por seu superior imediato, como por seus pares e também por seus subordinados (RABAGLIO, 2004). Este método, no entanto, não consegue avaliar de forma ampla uma pessoa, uma vez que os subordinados não têm plenas condições de dizer se os resultados atingidos pelo chefe estão alinhados com as expectativas dos superiores a este último. O fator “comportamento” parece ser dos poucos a terem plena cobertura por este método (DUARTE, 2001 e DUTRA et al., 2001). Mas de qualquer forma a nota final que o avaliado recebe é resultado de decisão humana. Se um grupo de pessoas tiver

interesses em beneficiar determinada pessoa, a tendência é que sejam atribuídas boas notas a esta pessoa, e a recíproca é verdadeira, deixando o avaliado à mercê de um processo com fortes influências subjetivas.

No início da década de 1990 com o advento do movimento chamado de “Reengenharia” (HAMMER, 1994) e o início do processo de Globalização dos Mercados, iniciou-se uma época em que entidades (pessoas, empresas e até países), anteriormente respaldados em algum tipo de esquema protecionista, começaram a receber o enfrentamento direto de outras, antes distantes e sem acesso, e a ter que brigar diretamente com esta concorrência. Problemas de eficiência começaram a aparecer. A estrutura hierárquica como elemento diferenciador da competência pessoal foi abalada, uma vez que, por exemplo, um gerente de uma unidade com determinado faturamento anual passou a agregar mais valor à organização do que o diretor de outra unidade com faturamento inferior (DUTRA et al., 2001). A obtenção de resultados passou a ser a palavra de ordem.

Neste contexto surge o modelo administrativo na área de Recursos Humanos chamado “Gestão por Competências”. O modelo teórico que embasa este movimento foi desenvolvido inicialmente por David McClelland na década de 1970 que concluiu que os testes tradicionais de conhecimento, aptidão e inteligência utilizados em seleção de pessoas não eram capazes de prever sucesso no trabalho e na vida cotidiana e sugeriu a introdução de outro elemento a ser medido que seriam as características pessoais do indivíduo que contribuíssem para o desempenho superior de uma tarefa específica. A essas características McClelland propôs o termo “competência” (DUARTE, 2001).

3.1 AVALIAÇÃO POR COMPETÊNCIAS

“O indivíduo é avaliado e analisado para efeitos de admissão, demissão, promoção, aumento salarial, etc. (...)” (DUTRA et al., 2001). Em função disso a definição de índices numéricos, ou conceituais, que sejam confiáveis é fundamental para que os processos de avaliação tenham maior eficácia. Indicadores errados levam a tomadas de decisão erradas. O modelo baseado em Competências oferece também melhores possibilidades de se definir indicadores mais precisos e, também, melhores possibilidades de se medir determinadas competências por meio de

mecanismos de avaliação automatizados, o que vem a minimizar a interferência humana na atribuição da nota ou conceito.

O modelo da “Competência”, no contexto dos Recursos Humanos e predominantemente adotado no Brasil, está embasado na combinação de três fatores intitulados, na língua Portuguesa, pela sigla CHA (no original inglês KSA = Knowledge, Skills and Attitudes: PARRY, 1996).

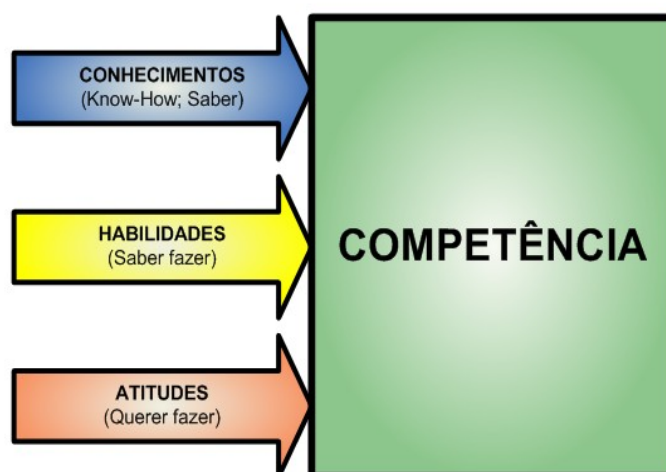


Figura 2 – Modelo CHA de “Competência”

Conhecimento sobre determinado assunto. *Know-how, Saber.*

Habilidade para produzir resultados com o conhecimento. *Saber Fazer.*

Attitude pró-ativa. Assertividade. *Querer Fazer.*

(RABAGLIO, 2004)

Segundo Dutra et al (2001), apesar deste modelo de avaliação de pessoas estar dominando o cenário corporativo dos departamentos de Recursos Humanos, por si não garante resultados satisfatórios, pois a realidade abrange outros fatores que o modelo CHA nem sempre contempla. A avaliação completa deve abranger quatro setores:

- TÉCNICO

- COMPORTAMENTAL
- RESULTADOS
- COMPLEXIDADE

O setor **técnico** se sobrepõe à componente **C** (conhecimento) do modelo CHA. Trata-se do que a pessoa de fato sabe a respeito da sua profissão e pode, em grande parte, ser averiguada por meio de testes escritos, exceção aos casos em que a profissão requeira alguma espécie de atividade física ou “desempenho” presencial, o que só pode ser comprovado mediante entrevista com especialistas no domínio.

O setor **comportamental** se sobrepõe à componente **A** (atitudes) do modelo CHA. São características pessoais que podem ser, também em grande parte, averiguadas via testes escritos. Na área de Recursos Humanos foram desenvolvidos testes para avaliação de características diversas desta natureza.

O setor **resultados** tem tanto a componente **H** (habilidades) do modelo CHA como a resultante do modelo de Avaliação por Objetivos, citada anteriormente. Em outras palavras, a pessoa não só tem que ter apresentado resultados satisfatórios, mas tem que demonstrar ter as habilidades para tal. Rabaglio (2001) ressalta a diferença entre “ser competente” e “ter competências”. O primeiro caso significa que determinada pessoa foi eficaz em uma determinada tarefa, mas não se pode garantir que o será da próxima vez. No segundo caso significa que determinada pessoa tem habilidades para transformar seus conhecimentos em resultados, compatíveis com determinado tipo de tarefa, o que a habilita a ser indicada para todas que se apresentarem no futuro, mas também não há como se garantir que esta pessoa logrará êxito apesar de toda a sua habilitação. A competência, neste quadrante, portanto, deve ser uma combinação de habilidades com resultados constatados (DUTRA et al, 2001). Resultados só podem ser constatados na prática, mas habilidades podem ser avaliadas, não só por constatações práticas, mas também por meio de provas situacionais que envolvem a simulação de situações reais típicas do cotidiano.

O setor **complexidade** abrange um tipo de habilidade (componente H do CHA) relacionada com o conhecimento tácito. Diz respeito à capacidade de uma pessoa em assumir responsabilidades e executar atribuições de maior

complexidade, o que vai se formando com o desenvolvimento e a experiência de vida da pessoa.

Neste modelo de Dutra et al. (2001), se somam os dois modelos mais utilizados, que é o modelo Americano, baseado em *inputs* (CHA) e o modelo Europeu, baseado em *outputs* (Resultados) (DUARTE, 2001).

Relembrando a definição de Scotty B. Parry, citada anteriormente, a Competência é conceituada como um agrupamento de conhecimentos, habilidades e atitudes. Em termos práticos, a Competência está associada à capacidade de efetivação de uma determinada tarefa que pode ser mais ampla ou mais focada em determinado item. Um gerente de departamento de informática, por exemplo, necessita ter uma ampla gama de conhecimentos e experiências para conduzir a contento sua equipe, não só técnicos, mas administrativos e de relacionamento interpessoal, ao passo que um programador júnior precisaria saber apenas a linguagem de programação respectiva ao seu projeto para poder levar a cabo a sua missão. O conjunto de itens CHA que compõe a Competência do gerente é bem mais amplo que o conjunto do programador júnior. Já os itens CHA vistos de forma isolada têm, em princípio, o mesmo significado tanto para o gerente como para o programador, como o conhecimento em um determinado idioma. Falar este idioma fluentemente independe da abrangência da Competência em que estiver inserido. Desta forma infere-se que os itens CHA são os elementos indivisíveis (ou atômicos) da Competência e a Competência o elemento variável em função da sua composição “atômica”.

Para se referir a estes itens CHA, será adotada, neste estudo, por falta de palavra adequada no idioma português, e também por sua larga utilização mundial, a palavra do idioma inglês “skill”. Portanto uma dada Competência será composta por SKILLS (em maiúsculas). Os padrões IMS [2002], IEEE [2005] e HR-XML [2007] modelam basicamente este SKILL. Para estes a palavra *competency* serve para designar tanto os *knowledgements* (conhecimentos), quanto os *skills* (aptidões de forma geral), quanto as *attitudes* (atitudes), quanto as *abilities* (habilidades). Portanto, no modelo deste estudo, onde uma Competência é um conjunto de SKILLS necessários para levar a cabo determinada tarefa, nestes padrões esta nossa Competência seria o conjunto de *competencies*.

Relembrando que o objetivo principal deste estudo é idealizar um modelo computacional para a Competência, de forma a se conseguir identificar com mais

precisão se determinada pessoa estaria apta a desempenhar determinada tarefa, o sucesso deste modelo dependerá do quão correta e precisa for a modelagem dos SKILLS. Outro fator é a relatividade do SKILL associada, não à tarefa em si, mas ao contexto da entidade interessada em medir a Competência de determinada pessoa. O significado de conhecimentos avançados em uma determinada linguagem de programação para uma fábrica de software multinacional pode se estender, desde os padrões de estruturação de programa utilizados mundialmente, até as diferenças entre versões da linguagem. Para uma consultoria de porte médio, especializada em determinado segmento de sistemas, o conhecimento avançado necessário talvez não seja tão amplo.

Possivelmente a distância entre contextos seja uma realidade que nunca deixe de existir. Uma pequena organização não teria eficiência de operação adotando os mesmos parâmetros de referência de uma grande corporação. De forma que um dos desafios é se conseguir chegar a um mecanismo aferidor de competências que também possa ser ajustado ao contexto de interesse.

Outro fator a considerar é a quantidade de SKILLS necessários para compor uma determinada competência. Bailey et. al. (2001) relaciona 85 SKILLS necessários aos programadores de computador que trabalham no segmento industrial norte americano, relação esta que se apresenta de forma ordenada pela importância do SKILL. Esta lista, e a ordem de importância dos itens, foi fruto de uma pesquisa de campo envolvendo entrevistas com gestores de diversos níveis na área de informática das indústrias. O SKILL número 1 da lista foi “Habilidade para ler, entender e modificar programas escritos por outros programadores”, com 96,89% de importância. Já o último colocado foi “Habilidade para ler e entender balanços e relatórios financeiros” com 2,76% de importância. Evidentemente, pela própria experiência prática no cotidiano das empresas, sabe-se que, mesmo considerando-se esses 85 SKILLS com suas respectivas relevâncias, dificilmente se conseguiria encontrar alguma pessoa que detenha todos. O caso do SKILL, último colocado, provavelmente só será encontrado em programadores que fazem ou já fizeram programas que gerem balanços e relatórios financeiros. Torna-se importante um critério de seleção dos SKILLS necessários para uma determinada função de programador. Caso outros SKILLS relacionados, embora não obrigatórios, estejam presentes numa determinada pessoa, pode-se entender que esta pessoa apresenta

qualificações acima da média, o que muitas vezes é utilizado para desempate entre candidatos a uma oportunidade de trabalho.

No artigo de Aken et. al. (2007) são relacionados 82 SKILLS referentes aos requisitos do mercado empregador de profissionais de Sistemas de Informação em relação aos SKILLS apresentados pelos alunos formados pelas escolas do segmento. Nesse estudo foi utilizada, entre outras, uma variável intitulada *skill gap*, que representa as diferenças de um determinado SKILL entre um valor esperado e um valor disponível ou realizado. Uma determinada entidade empregadora, por exemplo, que deseje encontrar programadores, dentro de determinada faixa salarial, que obtenham determinada nota mínima em um teste padrão de linguagem C++, mas que esteja se deparando com uma realidade mais pessimista, teria duas alternativas: a redução do grau de exigência ou a criação de uma estrutura de treinamento para preparar os profissionais com o nível desejado. Este conceito pode ser utilizado também no sentido oposto, o do profissional que procura saber o quanto seus SKILLS se distanciam dos níveis exigidos pelo mercado empregador. Este profissional poderia, com essa informação (*skill gap*), buscar o conhecimento necessário para melhorar suas chances de competir no mercado de trabalho.

3.2 PADRÕES ESTRUTURAIS PARA MODELAGEM DE COMPETÊNCIAS

Em linhas gerais os padrões IMS (2002), HR-XML (2007) e IEEE (2005) modelam como *competency* o que neste estudo foi chamado de SKILL. O IMS (2002) e o IEEE (2005), especificam 5 elementos obrigatórios:

1. **identificador** – um código único e universal suficiente para referenciar o SKILL no sistema.
2. **título** - um nome (curto) para o SKILL que faça sentido ao usuário humano.
3. **descrição** - também para utilização do usuário humano onde se tem, sob a forma de texto, maiores detalhes e explicações acerca do SKILL.
4. **definição** - onde constam os tipos de dados, valores padrões, máximos e mínimos, enfim o domínio de operação do referido SKILL.
5. **metadados** - onde constam os valores aferidos na instância de determinado SKILL.

Como estes elementos serão implementados fisicamente não é objetivo dos padrões, ficando a critério do implementador. Este esquema está representado na figura abaixo.

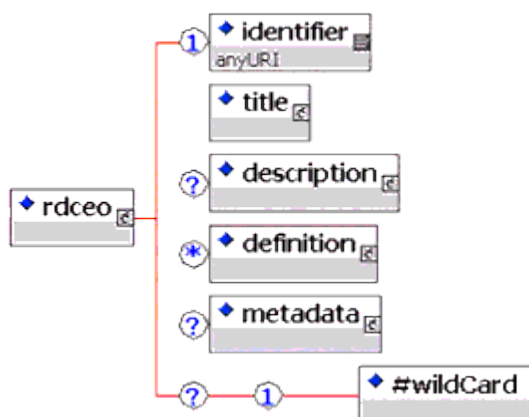


Figura 3 - esquema do Modelo da Competência no Padrão IMS (2002)

A figura 3, obtida do sítio da Internet do IMS-2002, representa o esquema da Competência pela sua designação genérica: RDCEO - *Reusable Definition of Competency or Educational Objective*. WildCard é um elemento de controle no padrão XML que não tem relação com o modelo da Competência em si.

Já o padrão HR-XML, com vistas ao escopo de Recursos Humanos, e onde este estudo se baseia com mais identidade, especifica maiores detalhes neste sentido. O esquema geral é visto na figura abaixo:

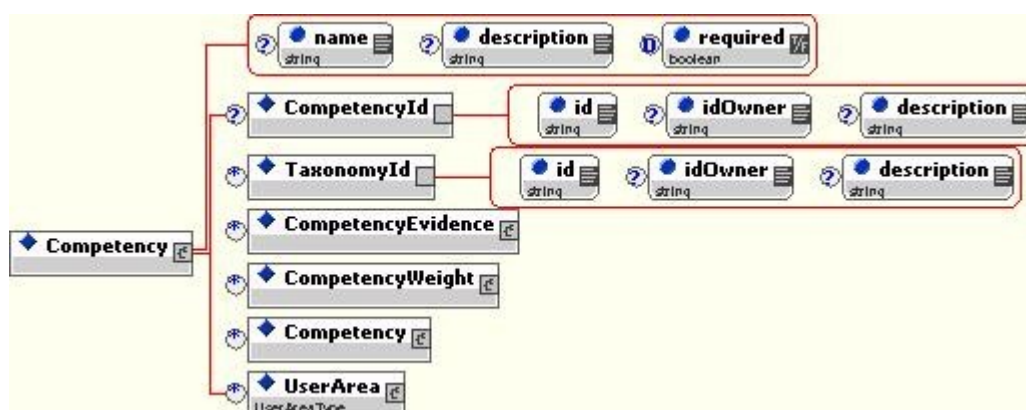


Figura 4 – Esquema do Modelo da Competência no Padrão HR-XML (2007)

Em relação ao padrão HR-XML (2007), temos as seguintes relações:

HR-XML (2007)	IMS (2002) / IEEE (2005)
Competency Um conhecimento, aptidão, habilidade e/ou outra característica relacionada, que seja específico, identificável, definível e mensurável, que uma pessoa possa ter e que seja necessária para levar a cabo a contento uma atividade dentro de determinado contexto.	Rdceo <i>Reusable Definition of Competency or Educational Objective</i>
CompetencyId	Identifier
Name	Title
Description	Description
CompetencyEvidence Armazena o conteúdo, valores numéricos ou textuais, que atestam a materialização de uma dada competência real. Pode-se ter desde notas (NAC) até relatórios e certificados.	Metadata
CompetencyWeight Representa a “nota” da Competency, geralmente um valor normalizado de 0 a 100.	(não especifica)

Tabela 1 – Termos comparativos entre os padrões HR-XML e IMS / IEEE

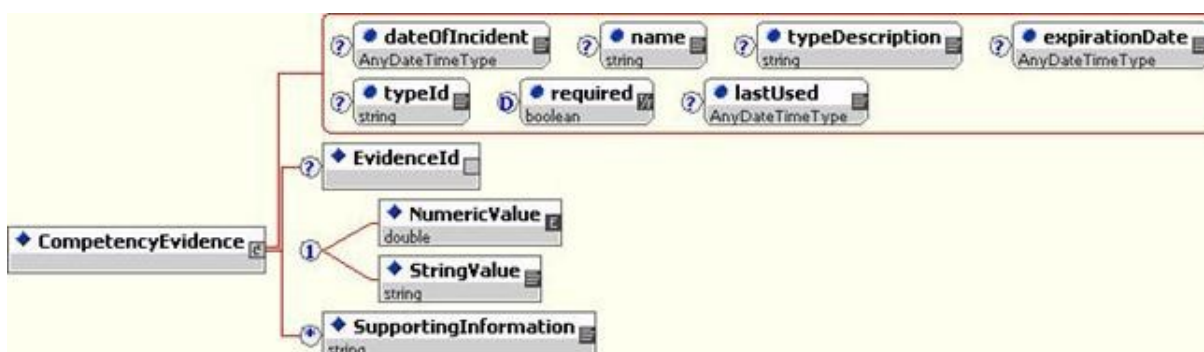


Figura 5 – Esquema do Modelo da Evidência da Competência no Padrão HR-XML (2007)

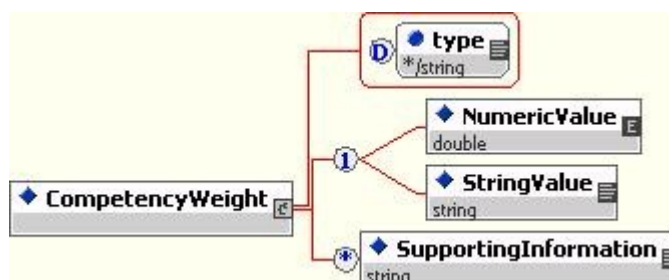


Figura 6 – Esquema do Peso da Competência no Padrão HR-XML (2007)

Para início da análise será considerando o SKILL composto por um bloco único de natureza identificadora (*CompetencyId*, *Name*, *Description*) ao qual está

atrelada uma subestrutura com dados diversos (*CompetencyEvidence*) relativos a este SKILL.

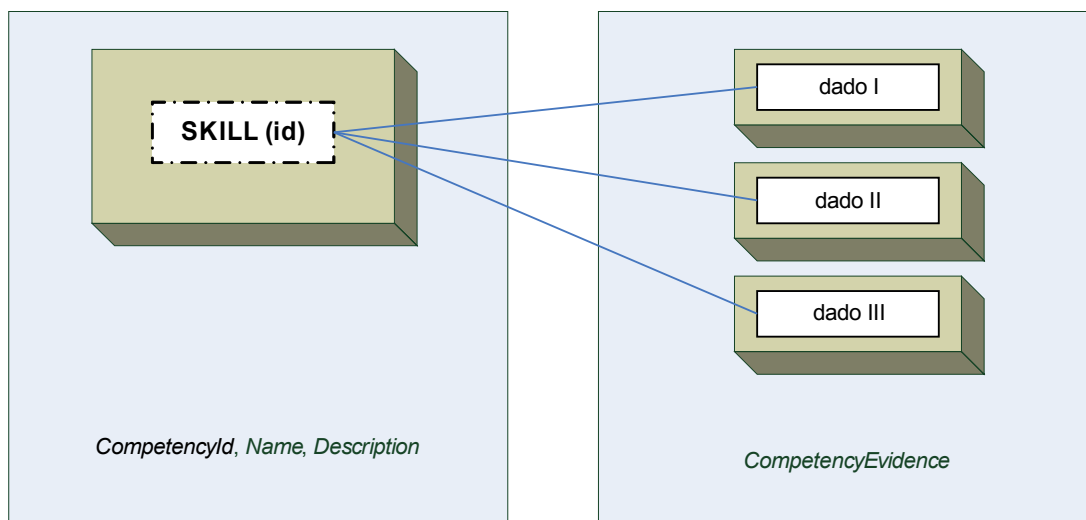


Figura 7 – Esquema simplificado do SKILL associado ao modelo HR-XML (2007)

Nesta subestrutura à direita, cada bloco intitulado “dado N” representa uma instância de *CompetencyEvidence* que, no contexto de um mesmo SKILL, devem ser diferenciadas pelo campo *EvidenceId* conforme apresentado na Figura 5.

No caso dos SKILLS que se representam por um único dado, o bloco “Dado I” seria o único componente desses SKILLS. Os blocos “Dado II” em diante seriam acrescentados à medida da necessidade de representação das informações do SKILL. Poder-se-ia ter também a possibilidade, no caso dos SKILLS compostos por mais de um item, de se apontar o dado mais representativo, ou de se armazenar um novo dado, resultado de algum tipo de cálculo resultante de todo o conjunto de dados, gerando uma informação consolidada, por exemplo.

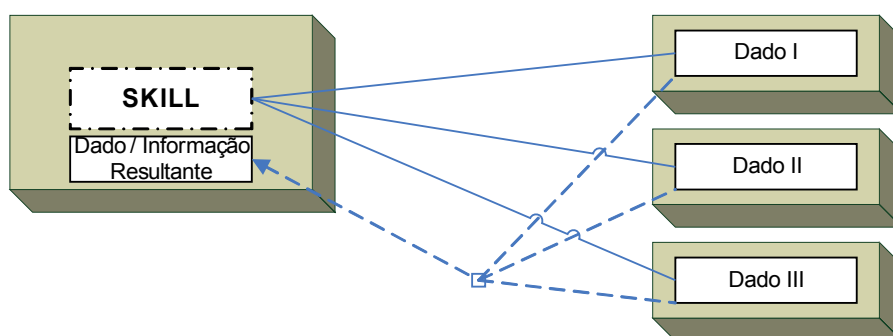


Figura 8 – Esquema simplificado do SKILL com informações adicionais

Como exemplo temos a seguir dois casos de SKILLs, sendo que o primeiro é um SKILL de estrutura mais simples e o segundo de estrutura um pouco mais complexa.

Exemplo 1: Certificação Microsoft em Windows Server 2003.

A Certificação é um tipo de diploma que determinado fabricante de software concede ao profissional que demonstrou proficiência em determinado produto. Neste exemplo o profissional se submeteu aos exames exigidos acerca do sistema operacional para servidores da Microsoft Corporation e foi considerado apto a prestar serviços neste produto dentro do padrão de qualidade mínimo exigido pelo fabricante. Este tipo de SKILL pode ter, por exemplo, dois tipos de dado, (1) Possui a certificação – SIM/NÃO e (2) No caso de possuir, em que ano foi feito o exame de certificação. Um terceiro dado possível, embora redundante em alguns casos, poderia ser um código que identificasse a instituição certificadora.

Exemplo 2: Conhecimentos no Idioma Inglês.

Os conhecimentos em determinado idioma podem ser medidos desde um simples indicador binário – Conhece: SIM/NÃO – até uma relação de pontos de medição dividindo os itens em: lê, entende, fala, escreve, e dentro destes se ter a classificação por níveis: básico, intermediário e avançado.

Idioma Inglês	BÁSICO	INTERMEDIÁRIO	AVANÇADO
(Informação I) Leitura	-	-	X
(Informação II) Escrita	-	X	-
(Informação III) Entendimento em conversação	X	-	-
(Informação IV) Fala	-	-	-

Tabela 2 – Exemplo de grade de requisitos para Conhecimentos no Idioma Inglês

No exemplo da tabela 2, o SKILL – Conhecimentos no Idioma Inglês – tem a seguinte composição: Leitura no nível avançado, Escrita no nível intermediário, Entendimento em conversação no nível básico e Fala = NÃO. Neste modelo cada item de informação tem três valores possíveis que possam caracterizar em que nível se encontra aquele determinado item do SKILL.

3.3 O ARMAZENAMENTO DE VALORES NO MODELO PROPOSTO DO SKILL

O modelo idealizado neste estudo pretende possibilitar a representação, em termos de valores, de qualquer tipo de SKILL. Para tanto um SKILL genérico deverá ter no mínimo um item de informação não havendo limite para o máximo de itens. A definição e modelagem de um dado SKILL é um trabalho que envolve a participação de no mínimo dois atores, um especialista técnico no modelo informatizado e um especialista no domínio do problema.

A evidência de um determinado SKILL (*CompetencyEvidence*) pode assumir formas diversas. A modelagem pretende representar o mais fielmente possível o objeto real. Segundo Seabra (2006), em seu estudo de modelagem para aplicações militares, o ciclo de vida do projeto de um modelo tem as seguintes etapas:

1. Definição do problema – parte fundamental onde uma definição incorreta leva fatalmente a um modelo incorreto.
2. Construção (idealização) do modelo – nesta etapa Seabra (2006) considera que a criatividade do analista construtor é de muita importância para que a materialização do modelo espelhe a realidade no mais alto grau possível. Segundo ele “Se o modelo não representar o sistema real com fidelidade, os resultados serão distorcidos”.
3. Solução do modelo – “algoritmo mais adequado em termos de rapidez de processamento e precisão de resposta”.
4. Etapas finais de implementação e testes.

Pelo roteiro de Seabra (2006), a etapa 1, definição do modelo do SKILL, envolveria definir que tipo de estrutura deveria representá-lo. No exemplo anterior (Exemplo 2) pode-se “criar” uma representação literal. Neste caso, o bloco de dados poderá conter uma letra: “B” para representar o nível básico, “I” para o Intermediário e “A” para o Avançado. A Informação Resultante poderia ser uma *string* alfanumérica com as quatro letras referente a cada informação (na ordem): AIBX.

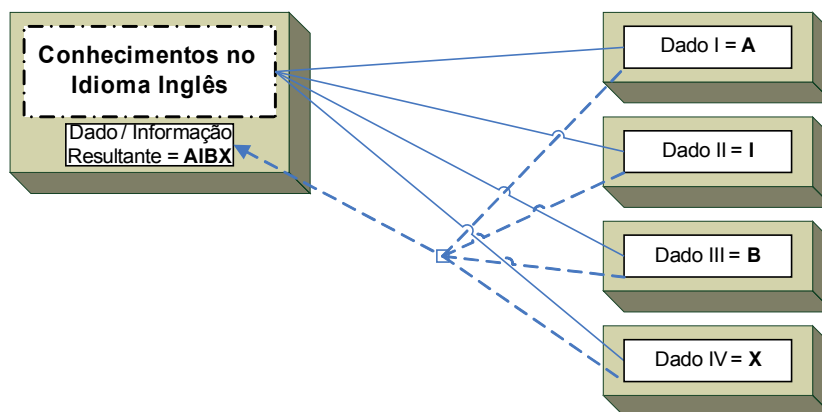


Figura 9 – Esquema simplificado do SKILL com informações adicionais valoradas

A representação também poderia ser numérica onde se atribuiria o valor 1 para o nível básico, 2 para o intermediário e 3 para o avançado e 0 para a não-existência. No exemplo anterior a Informação Resultante seria então 3210.

Um SKILL representativo da “Formação Acadêmica” poderia, por exemplo, se compor pelos seguintes itens:

1. Título do Curso
2. Instituição de Ensino
3. Ano de início
4. Ano de conclusão

Aqui haveriam também quatro instâncias de *CompetencyEvidence*.

Quanto ao “Nível de Aquisição do Conhecimento” de um aluno, que se compõe de um valor numérico que varia de 0 a 10 (LOCATELLI, 2007) associado a um conceito, pode-se ter os seguintes itens:

1. Conceito
2. NAC
3. Data

No sistema ACAVA, proposto por Locatelli (2007), a tabela de base de dados que armazena o NAC tem a seguinte estrutura:

1. codAlunoConc (informação do aluno)
2. codAluno (informação do aluno)
3. codConceito (informação do conceito)
4. nac (informação do conceito)
5. dataInc (informação do conceito)

No modelo proposto neste estudo, o armazenamento dos dados seguirá o mesmo conceito porém com uma estrutura física diferente visando possibilitar o registro de informações de qualquer natureza. A arquitetura funcional do sistema como um todo pode ser vista na figura a seguir.

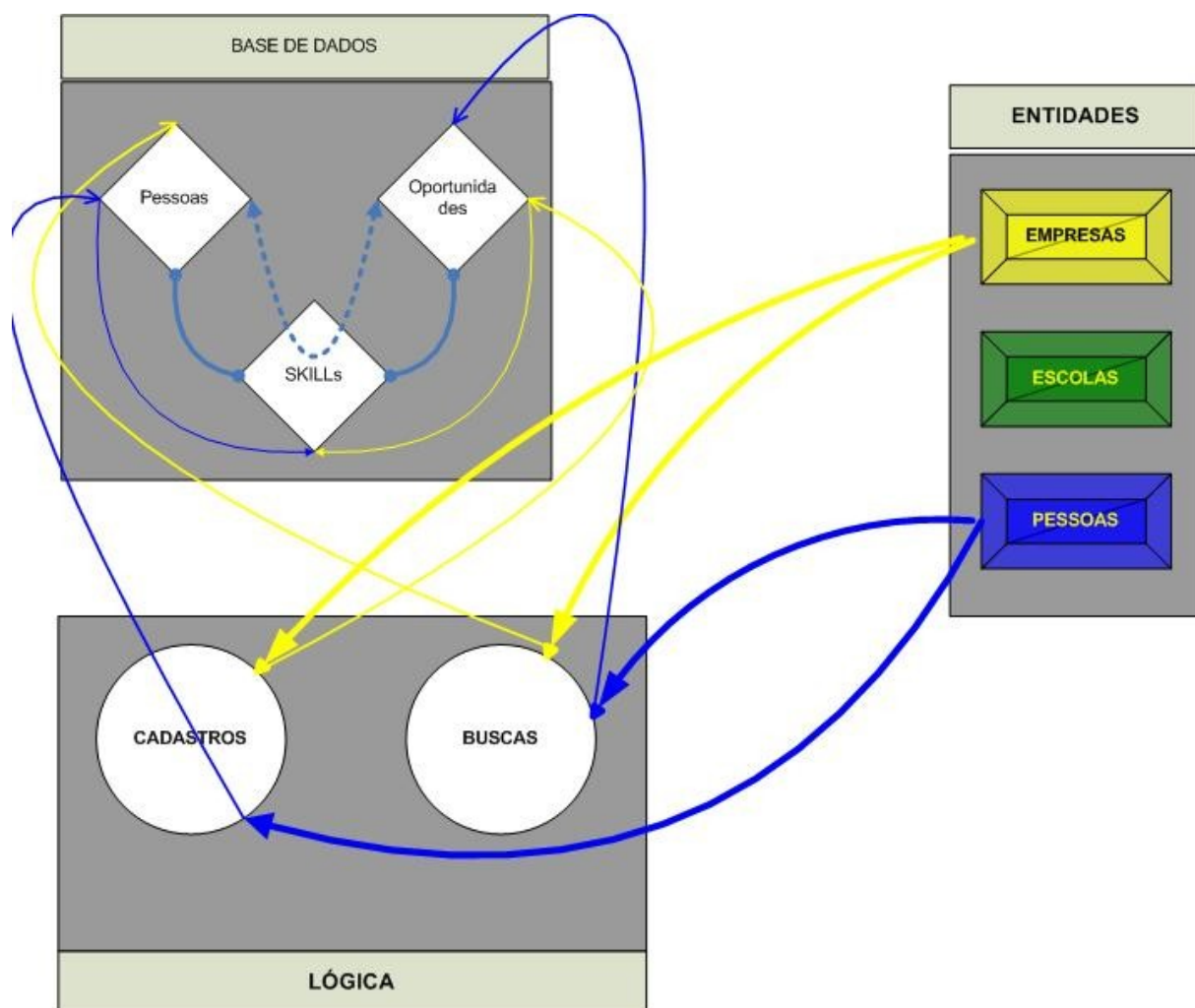


Figura 10 – Arquitetura funcional do modelo

4 MODELO COMPUTACIONAL PROPOSTO E TECNOLOGIAS NECESSÁRIAS.

Conforme apresentado no capítulo 1, o objetivo geral deste estudo é criar um modelo conceitual e respectivo modelo computacional de um ambiente de apoio à tomada de decisão na atividade de avaliação de pessoas por Competências.

Como visto nos capítulos anteriores, os padrões existentes intitulam de Competência o que neste estudo é um SKILL, ou seja, um item que compõe dada Competência. Tomando como exemplo o artigo de Bailey et. al. (2001), em que são relacionados 85 *skills* necessários para o bom desempenho da função de programador de sistemas de computador, esta competência, nos padrões (IEEE, 2005 e HR-XML, 2007), para que uma determinada pessoa apresente um bom desempenho na função de programador, são necessárias 85 “competências”, ao passo que no modelo deste estudo, baseado no modelo CHA, a Competência (no singular) necessária envolve 85 itens ou SKILLS. Em outras palavras “Quais as competências necessárias” versus “Que SKILLS compõe a Competência”. Trata-se apenas de uma diferença de classificação.

O padrão HR-XML implementa o modelo proposto neste estudo através do aninhamento de *competencies*, ou como chamado nesse padrão *Recursive Competencies* (vide ANEXO A).

Em termos práticos, por observação histórica, os sistemas computacionais desta natureza tem seus dados armazenados em bancos de dados e um programa (*software*), ou conjunto de programas, que manipula esses dados, localizando o que de interesse do usuário num dado momento e apresentando em algum tipo de dispositivo de saída, via de regra um monitor de LCD ou uma impressora, em um formato e/ou agrupamento que tenha um significado contextual. Em termos estruturais o sistema tem esses dois grandes blocos. A massa de dados das pessoas, empresas e instituições de ensino salvas no banco de dados e uma lógica de manipulação para alterações nos dados, busca, consolidação de informações e apresentação aos usuários.

Com base nestes conceitos e demais conceitos analisados anteriormente, chegou-se, após ensaios conceituais e experimentais diversos, ao modelo computacional descrito a seguir. Este modelo foi implementado experimentalmente sob a designação de SACHA (Sistema de Avaliação por Conhecimentos, Habilidades e Atitudes).

4.1 PLATAFORMA TECNOLÓGICA E AMBIENTE DE DESENVOLVIMENTO

Optou-se por uma plataforma que fosse *open source* e que fosse suportada por diversos sistemas operacionais. Para o bloco de *software* foi adotado o padrão **J2EE** (Java 2 Enterprise Edition - <http://java.sun.com/j2ee/overview.html>) de base Java, movido pelo *servlet container* **Tomcat** (<http://tomcat.apache.org>), e para o bloco de dados foi adotado o SGBDR (Sistema Gerenciador de Bancos de Dados Relacionais) **MySQL** (<http://www.mysql.com>), que atualmente está sob a tutela da Sun Microsystems (<http://www.sun.com/>), que mantém tanto a versão comercial quanto a versão *freeware*.

Dentro do modelo J2EE foi utilizada a estrutura MVC (Model-View-Control) em que tanto *Controllers* como *Views* foram implementados em scripts JSP (Java Server Pages) e a lógica de negócio e bibliotecas de utilitários em JavaBeans.

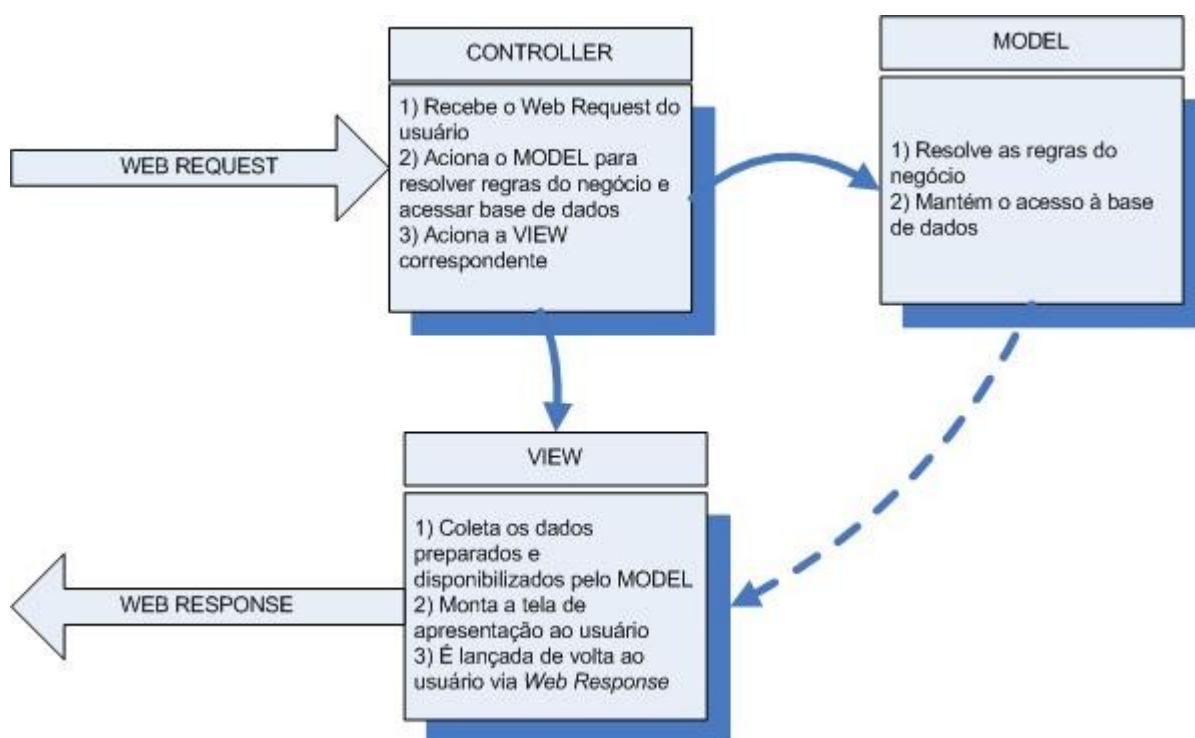


Figura 11 - Modelo MVC

A figura anterior mostra o diagrama da estrutura MVC, onde a *View* é representada pelos scripts que geram saída para o usuário e que permitem o retorno de comandos (*web requests*). Os *requests* gerados pelo usuários são recebidos

pelos *Controllers* que por sua vez tratam de providenciar os resultados esperados pelo usuário via lógica de negócios e manipulação nos estados dos *Models* e também definem a *View* a ser devolvida ao usuário. A *View*, para fechar o ciclo, antes de ser lançada no *web response*, lê os estados do *Model*, coleta os dados necessários e os dispõe no lay-out adequado para interpretação do usuário.

Quanto ao SGBDR escolhido, o MySQL, mesmo tratando-se de um banco de dados *open source* de grande utilização mundial, construiu-se o SACHA de forma a se utilizar de sintaxes SQL padronizadas ANSI 92 (<http://www.ansi.org>) de modo a poder operar em qualquer outro banco de dados que siga estes padrões sem que haja necessidade de reprogramação nas lógicas de controle.

O ambiente de desenvolvimento adotado foi o NetBeans da Sun Microsystems, versão 6.5; o ambiente de modelagem foi o DB Designer 4, também freeware, da fabForce.net e a máquina de desenvolvimento foi um Pentium Duo Core, T8100, 2.1GHz, Windows Vista profissional, memória 3 GB.

4.2 A BASE DE DADOS

O modelo deste estudo foi baseado na estrutura convencional de sistemas de bancos de dados, com as tecnologias adotadas, conforme ilustrado na figura abaixo:

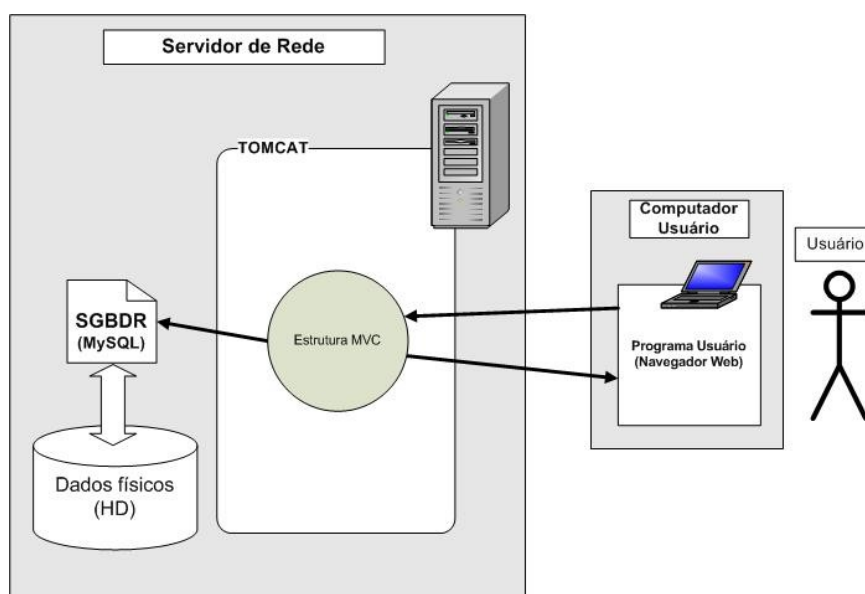


Figura 12 – Diagrama genérico de um sistema J2EE de banco de dados

Em linhas gerais deseja-se localizar, na base de dados, **Pessoas** que tenham determinados SKILLS que estejam em conformidade com os SKILLS solicitados por determinada **Oportunidade** (Competência) publicada por uma determinada Empresa. Sendo assim, necessita-se que nesta base existam as seguintes tabelas principais:

- Pessoas
- Oportunidades
- SKILLS
- Itens de SKILLS
- SKILLS das Pessoas
- SKILLS requeridos pelas Oportunidades
- Itens de SKILL das Pessoas
- Itens dos SKILL requeridos pelas Oportunidades

Vide Normalização deste modelo no APÊNDICE B.

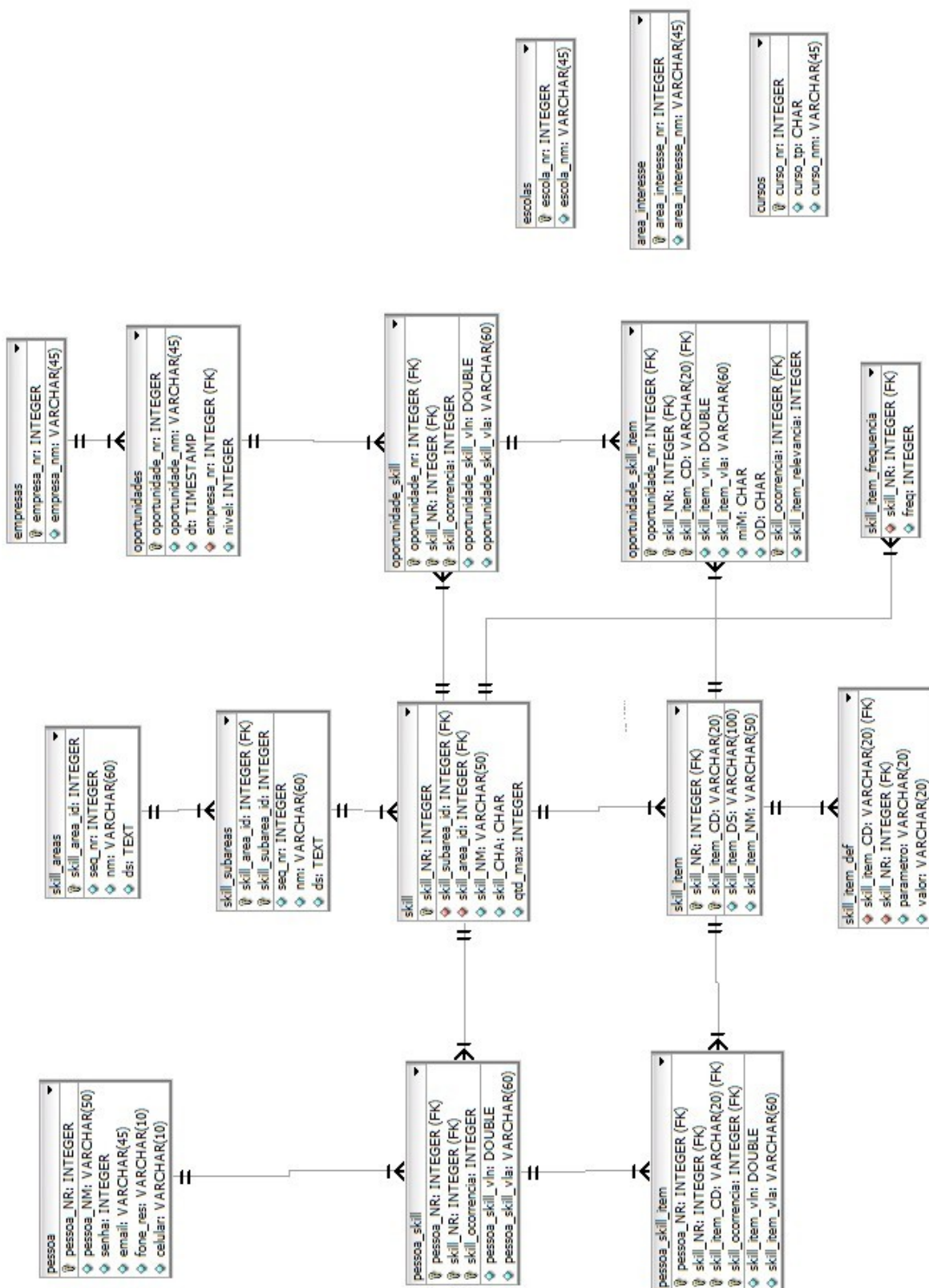


Figura 13 – Diagrama Entidade-Relacionamento do Projeto SACHA

4.2.1 O Modelo do SKILL

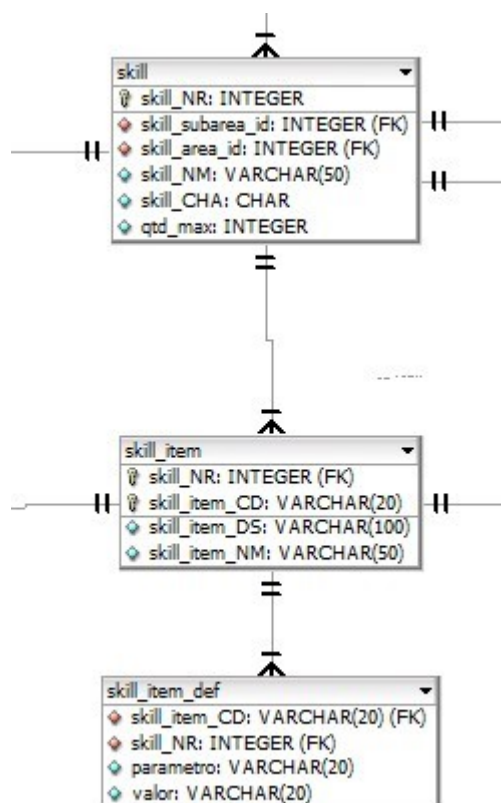


Figura 14 – Diagrama Entidade-Relacionamento do modelo do SKILL

A tabela **skill** relaciona os SKILLS cadastrados e reconhecidos pelo sistema.

Em relação ao padrão HR-XML:

HR-XML	SACHA
CompetencyId	skill_NR
Name	skill_NM
Description	Não implementado

Tabela 3 – Tabela “skill” segundo o padrão HR-XML

A tabela **skill_item** relaciona a declaração dos itens dos SKILLS cadastrados na cardinalidade de 1:N. Um SKILL pode se compor de quantos itens forem necessários para representar a realidade.

Em relação ao padrão HR-XML

HR-XML	SACHA
---	skill_NR = Chave Estrangeira
CompetencyId / id	skill_item_CD
Não previsto	skill_item_NM = nome do item
Description	skill_item_DS

Tabela 4 – Tabela “skill_item” segundo o padrão HR-XML

A tabela **skill_item_def** armazena as definições de um determinado SKILL que foi modelado, como será visto mais adiante no item 4.6, na chamada, neste estudo, de Série de Modelos 9000. Todas as definições então no binômio parâmetro=valor. Qualquer tipo de informação alfanumérica pode ser armazenada nesta tabela. Os parâmetros sempre estarão relacionados com os parâmetros previstos na classe que controla a lógica do referido SKILL.

Em relação ao padrão HR-XML

HR-XML	SACHA
---	skill_NR = Chave Estrangeira
---	skill_item_CD = Chave Estrangeira
CompetencyEvidence / typeld	parâmetro
CompetencyEvidence / StringValue	valor

Tabela 5 – Tabela “skill_item_def” segundo o padrão HR-XML

4.2.2 O Modelo da Pessoa e respectivos SKILLS

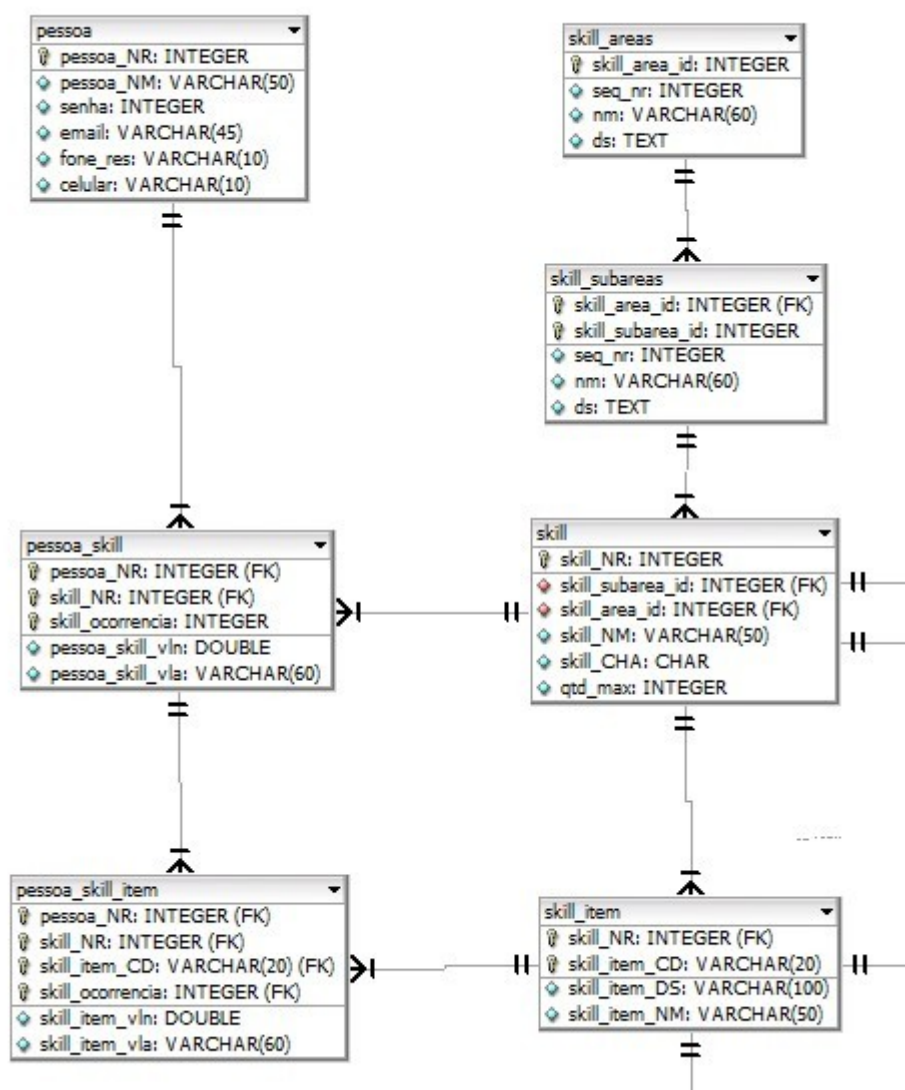


Figura 15 – Diagrama Entidade-Relacionamento do modelo “Pessoa”

A tabela **pessoa** relaciona todas as pessoas que se cadastraram no sistema e, portanto, visíveis aos mecanismos de busca. O modelo HR-XML não especifica o modelo da Pessoa em si já que esta se trata do “Competency Owner”. Nesta tabela, além do campo chave, pessoa_NR, devem constar as informações de identificação pessoal, que no caso deste estudo foram definidas apenas algumas para dar um mínimo de funcionalidade no protótipo implementado.

As tabelas **pessoa_skill** e a **pessoa_skill_item** relacionam, respectivamente, os SKILLS e respectivos itens de SKILL que determinada pessoa possui, acompanhados dos valores. Estas duas tabelas refletem exatamente o que foi ilustrado nas figuras 8 e 9 vistas anteriormente.

Em relação ao padrão HR-XML

HR-XML	SACHA
---	Pessoa_NR = Chave Estrangeira
---	skill_NR = Chave Estrangeira
---	skill_item_CD = Chave Estrangeira
---	skill_ocorrendia = (*)
CompetencyEvidency / NumericValue	peessoa_skill_vln
CompetencyEvidency / StringValue	peessoa_skill_vla
CompetencyEvidency / NumericValue	skill_item_vln
CompetencyEvidency / StringValue	skill-item_vla

Tabela 6 – Tabela “peessoa_skill_item” segundo o padrão HR-XML

(*) skill_ocorrendia é um campo não especificado de forma clara pelo padrão HR-XML. No modelo proposto por este estudo, possibilita a ocorrência de mais de um SKILL de mesmo identificador para uma mesma pessoa. O valor default é 1.

4.2.3 O Modelo da Oportunidade e respectivos SKILLS

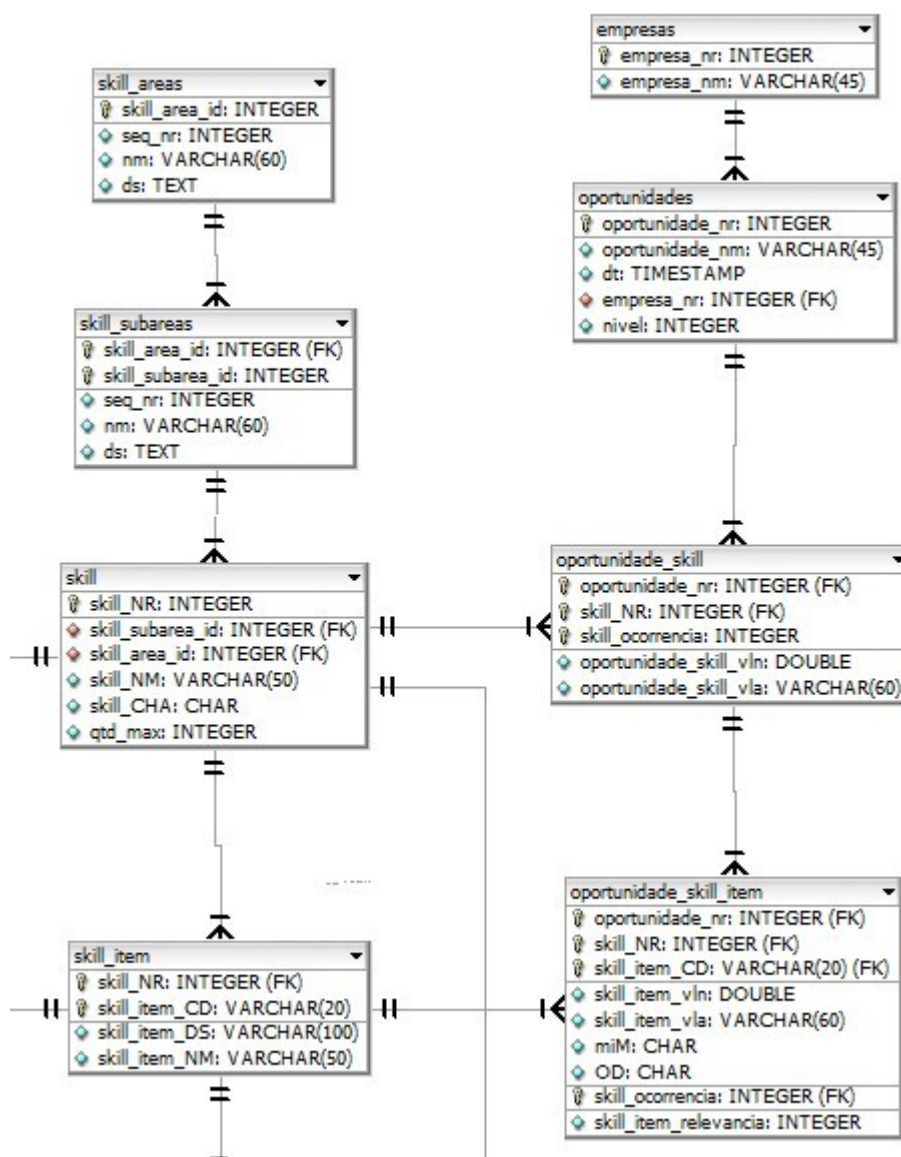


Figura 16 – Diagrama Entidade-Relacionamento do modelo “Oportunidade”

De forma análoga à Pessoa, a Oportunidade segue o mesmo modelo com alguns campos a mais.

A tabela **empresas** é uma tabela complementar para armazenar os geradores de oportunidades de trabalho cadastrados no sistema.

A tabela **oportunidades** é o par da tabela **pessoa**. Em termos comparativos, uma pessoa está para uma oportunidade. Os campos especificam os dados de identificação, que neste estudo foram escolhidos a data de publicação da Oportunidade, o nome ou título, por exemplo “Programador Java Júnior” e um indicador de nível hierárquico para possibilitar, no protótipo, que uma determinada pessoa procure oportunidades de um determinado nível.

A tabela **oportunidade_skill** é o par da tabela **pessoa_skill**, ou seja, os SKILLS relacionados em determinada Oportunidade são os procurados em alguma Pessoa que esteja cadastrada no sistema.

A tabela **oportunidade_skill_item** é o par da tabela **pessoa_skill_item** com alguns campos adicionais. Esses campos adicionais são os três descritos abaixo:

- **miM**: este campo, *data type* CHAR(1), especifica se o item do SKILL deverá ser procurado na base de dados como sendo valor “no mínimo” (**m**) em relação ao especificado no campo **skill_item_vln**, se “igual” (**i**) ou se “no máximo” (**M**). Em termos práticos, nos sites de Currículos convencionais, tem-se por padrão trazer no resultado da busca todas as pessoas que tenham no mínimo aquele valor ou aquela nota, entendendo-se que em se conseguindo recursos humanos melhores que o mínimo especificado, melhor para a organização contratante. Mas o modelo deste estudo prevê a possibilidade de que se possa fazer a busca com valores iguais ou máximos.
- **OD**: este campo, *data type* CHAR(1), especifica se o item do SKILL deverá ser obrigatório (**O**) ou desejável (**D**). Os itens obrigatórios excluem do resultado de busca as pessoas que não tiverem aquele determinado item ou não atenderem à especificação **miM**. Os itens desejáveis serão utilizados para compor a somatória adicional de pontos para gerar listas ordenadas na forma de *ranking*, dando mais recursos à empresa empregadora para identificar as melhores pessoas para preencher o posto.
- **skill_item_relevancia**: este campo, *data type* INTEGER, possibilita que a empresa empregadora pondere os itens, atribuindo mais peso a uns e menos a outros, de forma a ajustar a lista de *ranking* aos itens de SKILL que tem maior importância na situação. No protótipo implementado neste estudo o valor default é 10, podendo variar 0 a 10. Foi implementado um valor 15 com objetivo de possibilitar, uma vez configurada toda a Competência, dar destaque a determinado item de SKILL sem a necessidade de se proceder ao rebaixamento de todas as demais relevâncias.

O padrão HR-XML não especifica esses campos.

4.2.4 Tabelas de Referência

Todo SKILL e outros atributos em que há interesse na formação do perfil geral da pessoa são armazenados nas tabelas **pessoa_skill** e **pessoa_skill_item** com seus valores atribuídos, ou no campo **skill_item_vln** ou no **skill_item_vla**. Incluem-se, por exemplo, nos campos de valores, geralmente no campo **skill_item_vln**, o *id* de um curso que tenha sido realizado pela pessoa. Este *id* é proveniente da Chave Primária de uma tabela em que contenha “Cursos”, que no caso do protótipo deste estudo trata-se do campo **curso_nr**. No caso deste último exemplo não há como estabelecer um relacionamento de Chave Estrangeira no banco de dados, pois o campo **skill_item_vln** contém números das mais variadas origens inclusive “Chaves Estrangeiras” de diversas outras tabelas.

Essas tabelas, cuja Chave Primária é referenciada no campo **skill_item_vln**, e portanto não sofrem controle de integridade referencial exercido pelo SGBDR, mas que necessitam deste controle no nível da aplicação, foram chamadas, neste estudo, de Tabelas de Referência.

No protótipo implementado existem 3, **escolas**, para cadastro de entidades acadêmicas em que as pessoas possam obter graduação em determinados cursos, **area_interesse**, para cadastro das áreas em que pessoas possam estar interessadas em trabalhar e **cursos**, para cadastro de cursos existentes.

4.3 SQL APLICADA AO MODELO

Apesar da simplicidade do modelo DER, a sintaxe SQL necessária para localizar os SKILLS de interesse não é tão direta, sendo necessários alguns recursos de agrupamento e contagem para que se consiga identificar se determinada pessoa possui todos os SKILLS requeridos ou não.

Nas estruturas das bases de Currículos convencionais, em que cada SKILL é representado em uma coluna distinta na tabela da base de dados, os itens obrigatórios da Competência são buscados encadeando-se os argumentos pelo conector AND. No modelo deste estudo, em que os valores dos itens são armazenados na mesma coluna, este tipo de encadeamento não funciona, sendo possível apenas pelo conector OR associado à contagem de itens retornados.

No exemplo abaixo, a query SQL pretende recuperar registros na tabela `peessoa_skill_item` que tenha o SKILL de *CompetencyId* igual a 1 e itens de SKILL cujas *CompetencyId* sejam 'A' com valor 3, ou 'B' com valor 1, ou 'C' com valor 2, que tenha relação com a mesma pessoa (GROUP BY `peessoa_NR`) e que o agrupamento contenha 3 registros. Se essas condições não forem todas satisfeitas em uma determinada pessoa encontrada na base de dados, esta não será incluída no *ranking*.

```
SELECT COUNT(peessoa_NR),peessoa_NR
FROM peessoa_skill_item
WHERE
skill_NR=1 AND (
    (skill_item_CD='A' AND skill_item_vln=3) OR
    (skill_item_CD='B' AND skill_item_vln=1) OR
    (skill_item_CD='C' AND skill_item_vln=2) OR
)
GROUP BY peessoa_NR HAVING COUNT(peessoa_NR)=3
```

Neste primeiro protótipo não foi explorada a pesquisa pela tabela `peessoa_skill`, onde estaria armazenada a informação global do SKILL, mas sim pela tabela `peessoa_skill_item`, onde as possibilidades de refinamento por item são bem mais amplas.

4.3.1 A busca por múltiplos SKILLS – uma Competência

Localizar pessoas que possuam determinado SKILL, como visto, envolve a especificação valorada dos itens do SKILL, o que resulta em uma *query* composta e de relativa complexidade. Mas o objeto deste estudo é a Competência, entidade formada por uma composição de SKILLS. O exemplo da tabela 7 mostra uma Competência composta por 3 SKILLS.

SKILL 1 (Língua estrangeira)	Item 1 (leitura) = 3 (avançado) Item 2 (escrita) >= 1 (básico) Item 3 (compreensão) = 3 (avançado) Item 4 (fala) >= 1 (básico)
SKILL 2 (Linguagem de programação Java) 0 a 2,9 = pouco conhecimento 3 a 4,9 = conhecimento básico 5 a 6,9 = conhecimento intermediário 7 a 10 = conhecimento avançado	Item 1 (único) >= 7 (avançado)
SKILL 3 (Formação Superior) Engenharia = 15 ou Matemática = 68 ou Física = 37 Tempo de formado >= 5 anos	Item 1 (curso) = 15 ou 68 ou 37 Item 2 (tempo format.) → ano atual - ano formatura >= 5

Tabela 7 – Exemplo de formação de uma Competência no modelo computacional proposto

Neste exemplo procura-se pessoas que leiam e compreendam bem o idioma inglês, que tenham conhecimentos avançados na linguagem de programação Java e que tenham formação em Engenharia, Matemática ou Física há mais de 5 anos. O SKILL número 1 se compõe de 4 itens, o SKILL de *id* 2 é formado por uma única nota que define uma classificação entre 4 faixas e o SKILL da formação acadêmica, no exemplo, está cadastrado na base de dados como o curso de Engenharia tendo o *id* 15, o de Física o *id* 37 e o de Matemática o *id* 68. O segundo item deste SKILL de *id* 3 é o ano de formatura, sendo que para se obter o tempo de formado basta subtrair o ano atual do ano de formatura (a rigor seria necessário saber se o ano corrente encontra-se no início, meio ou fim para um cálculo mais preciso, mas esta precisão não vem ao caso neste momento do estudo).

Para que esta busca seja realizada na base de dados, a *query* SQL deverá ter a seguinte estrutura:

```
SELECT COUNT(pessoa_NR),pessoa_NR FROM pessoa_skill_item WHERE
```

(argumentos do primeiro SKILL)

```
(skill_NR=1 AND ((skill_item_CD='1' AND VALOR=3) OR (skill_item_CD='2' AND VALOR>=1) OR (skill_item_CD='3' AND VALOR=3) OR (skill_item_CD='4' AND VALOR>=1))
```

OR

(argumentos do segundo SKILL)

```
(skill_NR=2 AND skill_item_CD='1' AND VALOR>=7)
```

OR

(argumentos do terceiro SKILL)

```
(skill_NR=3 AND ((skill_item_CD='1' AND VALOR=15) OR (skill_item_CD='1' AND VALOR=68) OR (skill_item_CD='1' AND VALOR=37)) OR (skill_item_CD='2' AND "<ano atual>" - VALOR >= 5))
```

```
GROUP BY pessoa_NR HAVING COUNT(pessoa_NR) >= 7
```

É formado um bloco de argumentos para cada SKILL, também concatenados pelo operador OR (o operador AND só se aplica na identificação do registro: skill_NR + skill_item_CD + skill_item_vln). O operador HAVING COUNT deve especificar a quantidade mínima de registros esperados, que neste exemplo é de 7, podendo chegar a 9 caso a pessoa tenha formação acadêmica superior nos 3 cursos elencados na Competência-exemplo ilustrada na tabela 7.

Trata-se de uma *query* composta de 3 grupos de argumentos, os referentes ao SKILL número 1, os referentes ao número 2 e os do número 3 (supõe-se que os números de cadastro destes SKILLS sejam 1, 2 e 3). O primeiro grupo é idêntico ao analisado anteriormente, o segundo grupo é o mais simples exigindo apenas que o SKILL seja o de número 2 e que o item 1 tenha o valor maior ou igual a 7, e o terceiro grupo tem 3 alternativas para o item 1 e um valor calculado para o item 2.

O modelo deste estudo não pesquisou as conseqüências no desempenho do SGBDR nos casos de Competências compostas por grandes quantidades de SKILLS. A “Competência” de Bailey et al. (2001) em que são relacionados 85 SKILLS necessários aos programadores do mercado de trabalho norte-americano, por exemplo, geraria uma *String* SQL com 85 blocos de argumentos semelhantes aos do exemplo anterior. Mesmo que um determinado servidor, com determinada configuração de hardware, não tivesse problemas em executar esta *query*, não se poderia dizer o mesmo em uma situação de tráfego mais intenso em que o SGBDR precisaria interpretar e executar rapidamente as *queries* entrantes para evitar problemas de enfileiramento. No entanto, em termos práticos, inclusive pela própria vivência do autor em ambientes operacionais de recrutamento e seleção de pessoas, os requisitos obrigatórios raramente ultrapassam 6 ou 7. Os demais requisitos entram na categoria de “Desejáveis”, ou seja, não excluem as pessoas que não os possuem, mas as classificam melhor na lista de *ranking*. De forma que uma Competência composta por 10 SKILLS obrigatórios é bastante rara de ocorrer, mesmo porque o afunilamento que se forma à medida em que novos SKILLS são acrescentados acaba por reduzir demasiadamente a lista resultante de possíveis candidatos. Como será visto mais à frente, os SKILLS classificados como

“Desejáveis” não entram na query SQL, ficando o custo de processamento por conta do algoritmo implementado no modelo.

4.4 CASOS DE USO

No âmbito deste estudo, 3 são os atores e 6 são os casos de uso de interesse.

Atores:

- **Empregador:** entidade geradora de oportunidades de trabalho; define e configura as Competências desejadas.
- **Profissional:** Pessoa à procura de oportunidades de trabalho.
- **Aluno/estagiário:** Pessoa em formação acadêmica em busca da primeira oportunidade de trabalho.

Casos de uso:

- **Criar/alterar oportunidade + associar SKILLS:** o empregador cria uma nova oportunidade e seleciona os SKILLS necessários com as configurações desejadas.
- **Buscar Pessoas:** o empregador, de posse de uma oportunidade (Competência) previamente configurada, executa a rotina de busca, na base de dados, por pessoas que atendam aos requisitos obrigatórios da oportunidade em questão.
- **Incluir/Alterar cadastro Pessoal + associar SKILLS:** o profissional ou o aluno/estagiário cadastram-se no sistema, recebem uma senha, e selecionam os seus SKILLS com as especificações individuais.
- **Buscar oportunidades:** o profissional ou o aluno/estagiário executam a rotina de busca, na base de dados, por oportunidades publicadas das quais tenha possibilidades de se candidatar.
- **Comparar atributos pessoais X requisitos da oportunidade:** o profissional ou o aluno/estagiário executam a rotina que verifica em todos os itens se todos os seus SKILLS e demais atributos atendem aos requisitos de uma determinada oportunidade.

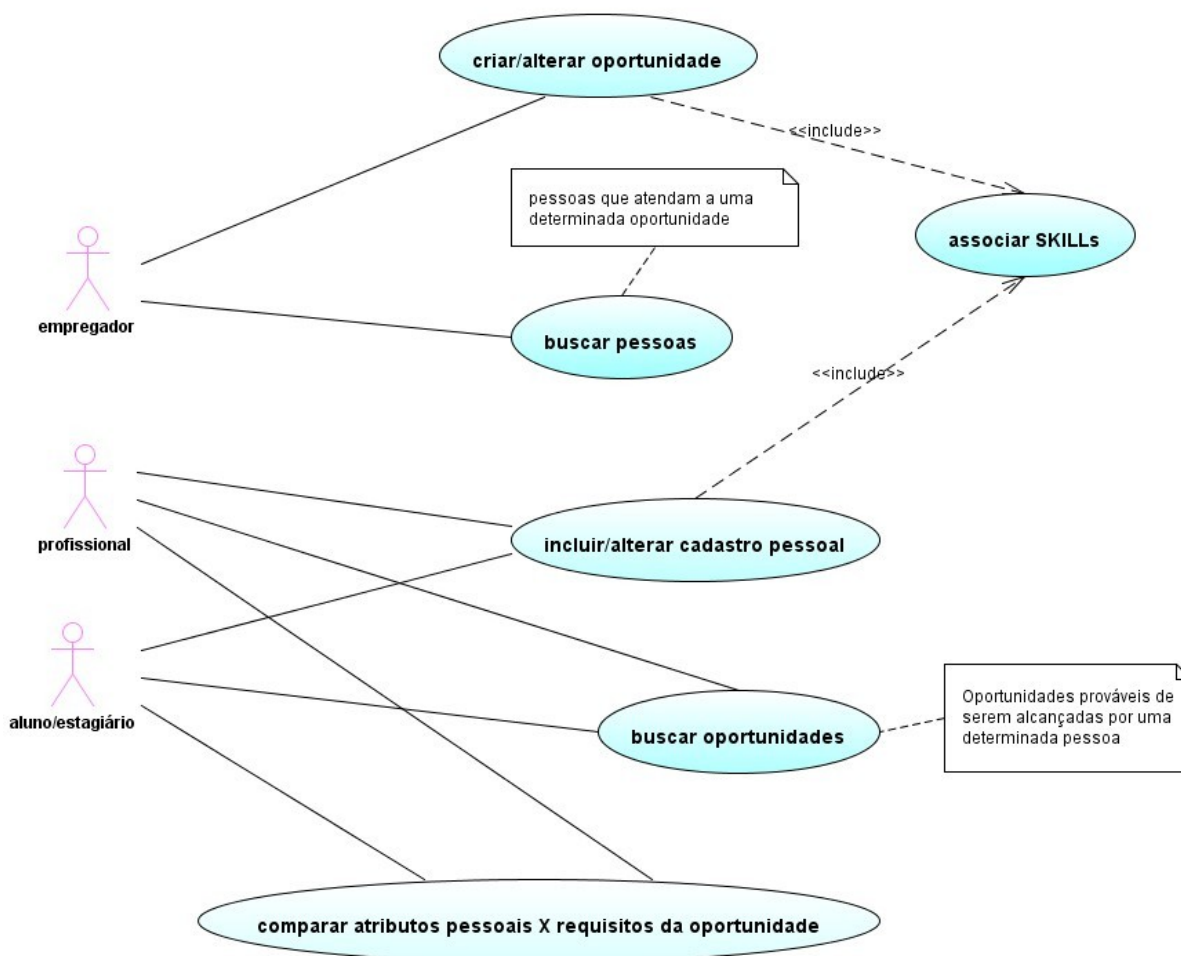


Figura 17 – Diagrama de Casos de Uso

4.5 DIAGRAMA DE CLASSES E OPERACIONALIZAÇÃO DO MODELO

Nas classes do SACHA existem 3 representações principais de objetos: A Pessoa, a Oportunidade e o SKILL. Todas as pessoas, assim como todas as oportunidades, tem idêntica estrutura em termos de classe. Os campos e métodos são os mesmos, assim como o corpo dos métodos também o é. De forma que para toda e qualquer pessoa, assim como para toda e qualquer oportunidade, uma classe para a pessoa e uma classe para a oportunidade são suficientes para representar esses dois domínios.

A representação do SKILL, no entanto, exige classes distintas para representação de SKILLS de naturezas distintas. Uma classe projetada para um SKILL de idioma estrangeiro, por exemplo, pode ser utilizada por outros SKILLS de

idiomas, desde que a representação seja a mesma. Já um SKILL de conhecimentos de linguagem de programação se representa por outros tipos de atributos, tendo que ter uma classe com outros tipos de campos e outros métodos.

Projetar uma classe diferente para cada tipo de SKILL pode ser algo inevitável pelas diferenças que existem entre esses tipos, mas inserir na programação do sistema todo um conjunto de instruções para operar com cada novo SKILL que surja no decorrer do tempo pode inviabilizar um sistema deste tipo em uma implementação produtiva. A solução encontrada foi definir os métodos necessários para todos os SKILLS e então implementar em uma classe pai os métodos de mesmo corpo e depois criar a classe específica de cada SKILL, todas seguindo uma estrutura de métodos de assinatura padronizada mas com os conteúdos específicos de cada caso. Além disso há uma interface, que toda classe de SKILL deve implementar, obrigando que a nova classe implemente os métodos exigidos. A função desta interface, bem mais do que padronizar os métodos dos SKILLS, é permitir que todo SKILL seja instanciado pelo nome desta interface, que é único, ao contrário das classes de cada SKILL que devem ter nomes distintos.

A figura 18 ilustra o diagrama de classes do SKILL.



Figura 18 – Diagrama de Classes do SKILL

No SACHA a definição de classes de SKILL se utilizou do prefixo **Skill** e o sufixo numérico de 4 dígitos, Sendo que a classe pai é a **Skill0000** e a partir daí podem ser criadas as classes **Skill0001** até **Skill9999**. A interface recebeu o nome

de **Skill**. Estas designações foram elaboradas com objetivo didático do projeto, pois em termos práticos pode-se ter qualquer nome para as classes e qualquer quantidade além das 9999 previstas na nomenclatura. No entanto quantidades desta monta não são gerenciáveis em implementações práticas, caracterizando a situação chamada “explosão de classes”, conforme discutido com mais detalhes no item 4.6.

Com esta estrutura a introdução de um novo SKILL no sistema deverá seguir o seguinte procedimento:

- 1 Criar a classe **Skill<nnnn>** (*)
- 2 Instalar o novo arquivo na pasta (*) `../webapps/sacha2009/WEB-INF/classes/sacha`
- 3 Relacionar o novo SKILL com a nova classe editando-se e recompilando-se a classe **_GetSkill()**.
- 4 Disponibilizar o novo SKILL para uso incluindo-o na tabela **skill** e respectivos itens na tabela **skill_item** (**).

(*) Caso não exista classe já implementada que atenda ao modelo

(**) No SACHA não foi implementada uma interface visual para esta operação

Abaixo está transcrita na íntegra a classe **_GetSkill()** no momento desta pesquisa:

```
public class _GetSkill {
    public _GetSkill() {
    }

    /**
    * Retorna a instancia da classe (objeto) correspondente correspondente ao identificador numerico do
    SKILL
    * @param hm
    * @return
    */
    public synchronized static sacha.Skill getSkill(HashMap hm) throws
    SQLException {

        sacha.Skill skill = null;

        int skill_NR = Integer.parseInt((String) hm.get("skill_NR"));

        switch (skill_NR) {
            /* Conhecimentos em idiomas
            case 1:
            case 7:
            case 8:
            case 9:
            case 10:
                skill = new sacha.Skill10001(hm);
                break;

            /* Conhecimentos de informatica nivel usuario
            case 2:
            case 3:
```

```

        case 4:
        case 5:
        case 6:
            skill = new sacha.Skill10002(hm);
            break;

        /* Naturalidade
        case 16:
        case 12: /* certificação em sistemas ABC (aproveitando mesma estrutura)
        case 13:
            skill = new sacha.Skill10016(hm);
            break;

        /* Template 9000
        case 19: /* Coeficiente de Inteligência
        case 20: /* Informática profissional - Linguagem Java
        case 21: /* Coeficiente de Concentração
            skill = new sacha.Skill19000(hm);
    }

    return skill;
}
}/* FIM DA CLASSE

```

Um objeto SKILL é instanciado via método `_GetSkill.getSkill(HashMap hm)`. O *HashMap* deve conter o campo identificado do SKILL (*CompetencyId*) o `skill_NR` e o campo `skill_ocorrencia` da seguinte forma:

```

HashMap hm = new HashMap();
hm.put("skill_NR", 3);
hm.put("skill_ocorrencia", 1);
sacha.Skill skill = sacha._GetSkill.getSkill(hm);

```

Este ultimo método localiza a classe, instancia e devolve o objeto SKILL.

Abaixo as classes Oportunidade e Pessoa.

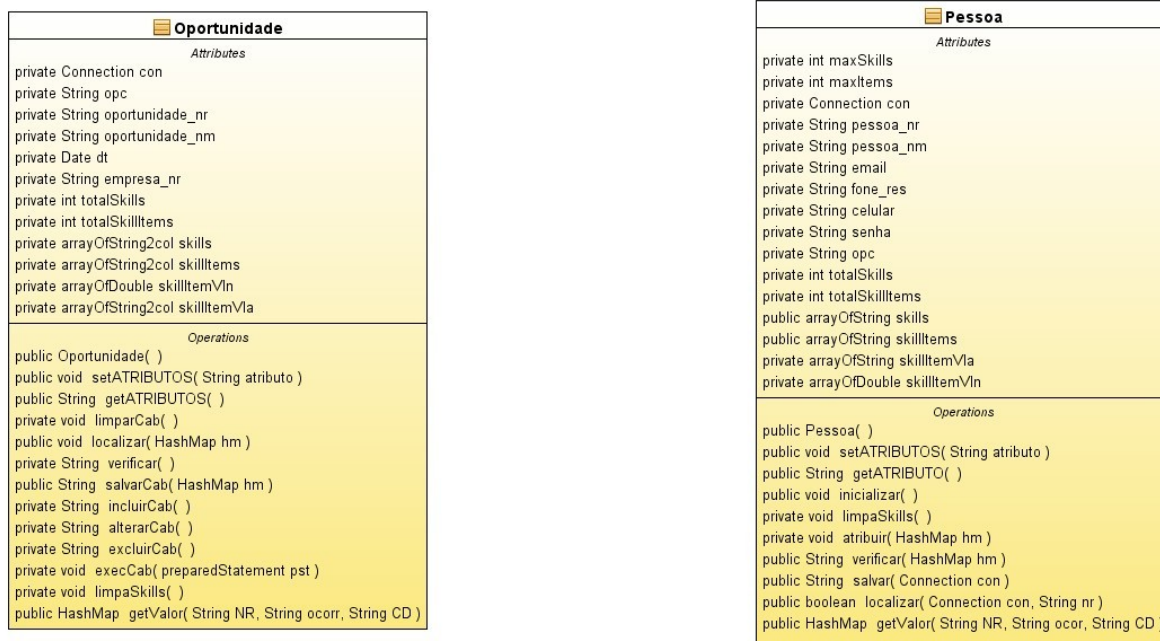


Figura 19 – Diagrama de Classes – Oportunidade e Pessoa

O relacionamento destas duas classes com a classe Skill0000 se dá, do lado do SKILL, pelos campos **pessoa_NR** ou pelo **oportunidade_nr**. O campo **POX** sinaliza se o SKILL foi instanciado por uma oportunidade, por uma pessoa ou de forma isolada:

- P – pessoa
- O – oportunidade
- X – sem vínculo

Nas classes Oportunidades e Pessoas, os respectivos SKILLS são relacionados nas arrays **skills** e **skillItems** e valores nas arrays **skillItemVln** e **skillItemVla**.

4.6 – A QUESTÃO DA EXPLOSÃO DE CLASSES E A SÉRIE 9000.

No modelo proposto, a par da escalabilidade oferecida pela estrutura da herança, existe a possibilidade de ocorrência da situação chamada “explosão de classes”, que se caracteriza quando a quantidade de classes-filhas aumenta a tal ponto que, não só o controle de catalogação, mas qualquer necessidade de manutenção no modelo pode demonstrar complexidade se o código a ser alterado

estiver na classe-filha. Todas as classes deverão sofrer as modificações necessárias, recompiladas e testadas.

Como já visto, não há necessidade que seja implementada uma classe para cada SKILL, mas pelo menos uma para cada tipo estrutural e comportamental de SKILL. A exemplo do conhecimento em línguas estrangeiras, a mesma classe pode ser utilizada para qualquer idioma, seja inglês, seja mandarim, mas mesmo assim a quantidade de tipos estruturais, embora não tenha sido feito este levantamento, estima-se não ser pequena.

Mediante parametrização externa à classe, pode-se obter mudanças de comportamento, o que possibilitaria a eliminação de classes de mesma estrutura mas com comportamentos diferentes.

Neste estudo foi implementada uma classe de nome Skill9000, e batizada de SKILL da série 9000, que implementa um SKILL de item único que possibilita o armazenamento de um valor numérico qualquer. A esta classe foi dada a funcionalidade de buscar parâmetros na base de dados em função do identificador (CompetencyId) do SKILL. Neste primeiro protótipo foram definidos quatro parâmetros:

1. Máximo
2. Mínimo
3. Escala composta por valores discretos ou contínuos
4. Escala na ordem direta ou reversa

Com estas informações a classe é capaz de, dado um valor, calcular o *Skill Score* utilizando interpolação linear, possibilitando que qualquer SKILL representado por um valor numérico dentro de uma escala genérica, possa ser representado sem a necessidade de implementação de uma classe específica.

A abrangência de operação neste tipo de projeto de classe fica por conta do implementador. Por exemplo, à classe Skill9000, descrita acima, poderiam ser acrescentados mais dois parâmetros: se a interpolação é do tipo linear, exponencial, polinomial etc. e um índice que reflita de alguma forma o grau de curvatura da curva.

Com este tipo de estrutura consegue-se reduzir a quantidade de classes necessárias para representar os diversos tipos de estrutura e modos de comportamento dos SKILLS.

Há também a possibilidade de se inserir em uma mesma classe SKILLS semelhantes que tenham poucas diferenças de estrutura e operação mediante a inclusão de um parâmetro “tipo” e a inserção de condicionais **if (tipo == 1)** ou classificadores **switch (tipo)** nos trechos do código em que ocorre a variação, de forma que um trabalho de aprimoramento ao longo do tempo acabará por restringir a quantidade de classes a um número razoável.

No entanto a estrutura permite que um novo SKILL seja inserido através de uma nova classe. Casos em que um novo SKILL realmente não se encaixe em nenhuma classe já desenvolvida, poderá ser implementado separadamente até que se consiga fazer um trabalho de inclusão nas classes atuais.

O foco na parametrização deve ser uma preocupação constante no projeto dos tipos de SKILL.

4.7 OS CASOS DE USO DE “CADASTRAMENTO”

O cadastramento de uma Oportunidade, como o de uma Pessoa, seguem o mesmo padrão. Um cabeçalho de identificação da Oportunidade, assim como um cabeçalho de identificação da Pessoa. À partir do cadastramento desta identificação, se disponibiliza o hyperlink para inclusão de SKILLS e ATRIBUTOS.

Os ATRIBUTOS são parâmetros de interesse em buscas, mas que não são caracterizados como SKILLS, como por exemplo a data de nascimento.


		Sistema de Avaliação por Conhecimentos-Habilidades-Atitudes *** Baseado em Gestão por Competências ***	Cadastro de Oportunidades * incluir/alterar *
VOLTAR INDICE			
ALTERAÇÃO: REGISTRO ORIGINAL			
EMPRESA	ABC SERVIÇOS LTDA		
TÍTULO A SER PUBLICADO	Analista-Programador Java		
NÍVEL HIERÁRQUICO	analista/consultor		
DATA DA PUBLICAÇÃO	21/04/2009		
<input type="button" value="Enviar"/> <input type="button" value="Limpar"/> <input type="checkbox"/> Excluir esta Oportunidade (não pede confirmação)			
SKILLS & ATRIBUTOS incluir			
Atributos Pessoais			
--- Classificação Única ---			
Naturalidade	<ul style="list-style-type: none"> • Data de Nascimento: 01/01/1974: no mínimo: obrigatorio: 7 • Local de Nascimento: X 		alterar
Conhecimentos em Idiomas			
--- Idiomas Estrangeiros ---			
idioma - INGLES	<ul style="list-style-type: none"> • Lê: Conhecimento AVANÇADO: no mínimo: obrigatorio: 10 • Escreve: Conhecimento INTERMEDIÁRIO: no mínimo: obrigatorio: 10 • Fala/Entende: Conhecimento AVANÇADO: no mínimo: obrigatorio: 10 		alterar
Conhecimentos em Tecnologia da Informação			
--- Conhecimentos de Profissional ---			
Informática Profissional - Programação Java	<ul style="list-style-type: none"> • Valor: 5.0: no mínimo: obrigatorio: 10 		alterar
Concluído Internet Modo Protegido: Ativado 100%			

Figura 20 – SACHA: Tela Cadastro de Oportunidades

Os quadros informativos referentes a cada SKILL, compostos de 3 colunas (Figura 20), são fornecidos pelo método **skill.getHtmlDisplay()**. Este método existe na classe pai e nas classes específicas de cada SKILL. As classes específicas de cada SKILL geram o conteúdo da coluna do meio, sendo que todo o resto é produzido pelo método da classe pai. Em se tratando de uma oportunidade, como no caso do exemplo, o método vai buscar os dados na tabela **oportunidade_skill_item**, e se tratando de uma pessoa, os dados serão recuperados da tabela **pessoa_skill_item**. A seguir uma tela exemplo para o Cadastro de Pessoas.

Sistema de Avaliação por Conhecimentos-Habilidades-Atitudes
*** Baseado em Gestão por Competências ***

Cadastro de Pessoas
* Incluir/alterar *

VOLTAR | ÍNDICE

ALTERAÇÃO: REGISTRO ORIGINAL

NÚMERO DE CADASTRO: 201

NOME	PESSO_A_201
E-MAIL	
FONE RESIDENCIAL	
CELULAR	
SENHA	•

Enviar Limpar

SKILLS & ATRIBUTOS
[incluir](#)

Atributos Pessoais

— Classificação Única —

Naturalidade	• Data de Nascimento: 05/03/1978	alterar
--------------	----------------------------------	-------------------------

Formação Acadêmica

— Formação Superior —

Conhecimentos em Idiomas

— Idiomas Estrangeiros —

idioma - INGLES	<ul style="list-style-type: none"> • Lê: Conhecimento INTERMEDIÁRIO • Escreve: Conhecimento BÁSICO • Fala/Entende: Conhecimento BÁSICO 	alterar
-----------------	--	-------------------------

Conhecimentos em Tecnologia da Informação

— Conhecimentos de Usuário —

informatica usuario - EXCEL	• Grau de Conhecimento >>: Conhecimento INTERMEDIÁRIO	alterar
informatica usuario - WORD	• Grau de Conhecimento >>: Conhecimento INTERMEDIÁRIO	alterar

Internet | Modo Protegido: Ativado 90%

Figura 21 – SACHA: Tela Cadastro de Pessoas

A única diferença entre os dois cadastros, em relação à área de SKILLS e ATRIBUTOS, é que no Cadastro de Oportunidades são inseridas 3 informações a mais: (1) se a graduação deve ser considerada pela exatidão, pelo máximo ou pelo mínimo, (2) se o SKILL é obrigatório ou desejável e (3) qual o grau de relevância.

Vide os dois exemplos abaixo para alteração do SKILL “Idioma Inglês” no requisito da Oportunidade (Figura 22) e no SKILL da Pessoa (Figura 23). Ambas as saídas HTML são produzidas pelo método `skill.getHtmlIA()`.

SACHA: SACHA - Internet Explorer fornecido por Dell

http://localhost:17660/SACHA/pessoas\$oportunities_skill_IA.jsp?opc=A&oportunidade_nr=2&skill_NR=1&skill_ocorrencia=1

SACHA: SACHA

Sistema de Avaliação por Conhecimentos-Habilidades-Atitudes
*** Baseado em Gestão por Competências ***

idioma - INGLES

[CANCELAR](#)

Lê	<input type="radio"/> Não conhece	<input type="radio"/> Conhecimento BÁSICO	<input type="radio"/> Conhecimento INTERMEDIÁRIO	<input checked="" type="radio"/> Conhecimento AVANÇADO
Escreve	<input type="radio"/> Não conhece	<input type="radio"/> Conhecimento BÁSICO	<input checked="" type="radio"/> Conhecimento INTERMEDIÁRIO	<input type="radio"/> Conhecimento AVANÇADO
Fala/Entende	<input type="radio"/> Não conhece	<input type="radio"/> Conhecimento BÁSICO	<input type="radio"/> Conhecimento INTERMEDIÁRIO	<input checked="" type="radio"/> Conhecimento AVANÇADO

Enviar Limpar Excluir este SKILL (não pede confirmação)

Figura 22 – SACHA: Tela Cadastro de Oportunidades – alteração de SKILL

SACHA: SACHA - Internet Explorer fornecido por Dell

http://localhost:17660/SACHA/pessoas\$oportunities_skill_IA.jsp?opc=A&pesoa_NR=201&skill_NR=1&skill_ocorrencia=1

SACHA: SACHA

Sistema de Avaliação por Conhecimentos-Habilidades-Atitudes
*** Baseado em Gestão por Competências ***

idioma - INGLES

[CANCELAR](#)

Lê	<input type="radio"/> Não conhece	<input type="radio"/> Conhecimento BÁSICO	<input checked="" type="radio"/> Conhecimento INTERMEDIÁRIO	<input type="radio"/> Conhecimento AVANÇADO
Escreve	<input type="radio"/> Não conhece	<input checked="" type="radio"/> Conhecimento BÁSICO	<input type="radio"/> Conhecimento INTERMEDIÁRIO	<input type="radio"/> Conhecimento AVANÇADO
Fala/Entende	<input type="radio"/> Não conhece	<input checked="" type="radio"/> Conhecimento BÁSICO	<input type="radio"/> Conhecimento INTERMEDIÁRIO	<input type="radio"/> Conhecimento AVANÇADO

Enviar Limpar Excluir este SKILL (não pede confirmação)

Figura 23 – SACHA: Tela Cadastro de Pessoas – alteração de SKILL

4.8 OS CASOS DE USO DE “BUSCA”

As buscas se dividem em dois grandes grupos:

- Empresas procurando pessoas para trabalharem em seus projetos e necessidades
- Pessoas procurando oportunidades oferecidas pelas empresas

Como, na vida real, não é comum se encontrar uma determinada pessoa com as exatas características desejadas, tanto em tipo quanto em quantidade, ou uma pessoa encontrar uma oportunidade cujos requisitos estejam em exata conformidade com os seus SKILLS, se faz necessária a introdução de uma escala de valores para que se possa medir o quanto os SKILLS de uma determinada pessoa se distanciam dos especificados na Oportunidade.

Para que uma medição de valores seja possível, é necessário que o SKILL gere um valor numérico. Intuitivamente os SKILLS já apresentam um valor, mesmo que subjetivo, como o exemplo do conhecimento que é aferido em 3 níveis: básico, intermediário e avançado, ou o exemplo da habilidade que é avaliada em tempo ou quantidade de experiência como horas de vôo, número de saltos de pára-quedas ou anos de carreira.

Os valores como entendidos pelas pessoas, os ditos valores “comerciais”, se fazem necessários para apresentação ao usuário, mas para cômputo do valor total esperado por determinada Oportunidade se faz necessária a transformação deste valores “comerciais” em um valor normalizado.

No SACHA foi adotado o valor normalizado de 0 a 100 para cada SKILL (*CompetencyWeight*). Este valor é obtido pelo método **skill.getSkillScore()** que tem implementação específica para cada SKILL. Esta implementação é um dos pontos críticos no projeto de um novo SKILL pois a ponderação dos valores intermediários pode seguir diversos tipos de interpolação: linear, exponencial, logarítmica etc. A escolha do modelo que melhor represente a realidade pode não ser tão simples e a adoção de um algoritmo inadequado ocasionará distorções nos valores resultantes em relação ao real valor da pessoa (SEABRA, 2006). No exemplo dos níveis de conhecimento básico, intermediário e avançado com a interpolação linear se teria os seguintes *score*:

- Básico = 33
- Intermediário = 66
- Avançado = 100

Com uma interpolação exponencial o resultado já poderia ser diferente:

- Básico = 15
- Intermediário = 50
- Avançado = 100

Considerando o nível intermediário deste exemplo, se tem uma diferença de 16 pontos entre os dois critérios. Na escala de 0 a 100 seriam 16%, o que poderia, dependendo do caso, significar uma margem de erro elevada.

Com base no artigo de Aken et. al. (2007), foi introduzido no SACHA o conceito do *Skill Gap*, que representa a distância relativa ou percentual que está a somatória de pontos dos SKILLS da Oportunidade em relação à somatória obtida pelos SKILLS da pessoa no domínio da oportunidade. A esta somatória foi adotada, neste estudo, a expressão *Skill Score*.

Tanto o *Skill Gap* quanto o *Skill Score* tem duas categorias de valores: no âmbito do SKILL e no âmbito da oportunidade. A fórmula adotada para cálculo do *Skill Gap* é a seguinte:

SG = Skill Gap

SS = Skill Score

$$SG = \frac{SS_{pessoa}}{SS_{oportunidade}} - 1$$

No exemplo da figura 24 há uma oportunidade composta por quatro SKILLS, sendo dois obrigatórios, Idioma Inglês com *Skill Score* mínimo de 66 e Conhecimentos em Java, com *Skill Score* mínimo de 80, e dois desejáveis, Idioma Alemão com *Skill Score* mínimo desejável de 33 e Trabalho em Grupo com *Skill Score* mínimo desejável de 90.

A query SLQ gerada para busca na base de dados é montada com os blocos referentes a cada SKILL obrigatório, fornecidos pelo método **skill.getSql()** de cada um respectivamente. Só serão retornadas as pessoas que possuem os SKILLS obrigatórios com a pontuação “comercial” exigida.

Este *Result Set* é submetido então à rotina de cálculo do *Skill Score* e gerada uma *array* de *Strings* cujos itens tem o seguinte formato:

OOOOO-DDDDD-PPP

Onde:

- **OOOOO**: sequência *String* correspondente ao *Skill Score* **obrigatório** preenchida com zeros à esquerda para garantir a quantidade de 5 caracteres.
- **DDDDD**: sequência *String* correspondente ao *Skill Score* **desejável** preenchida com zeros à esquerda para garantir a quantidade de 5 caracteres.
- **PPP**: Número *id* de cadastro da pessoa na base de dados.

À esta *array* é aplicado o método *sort*, que segundo a documentação Java trata-se de um *merge sort* modificado com garantia de desempenho $n \cdot \log(n)$. A pessoa que obteve maior pontuação fica posicionada no final da *array*. Percorrendo-se a *array* no sentido inverso, obtem-se a lista em forma de *ranking*.

Caso a tela apresentada ao usuário só contenha o número *id* da pessoa, seria suficiente criar um hyperlink em cada um para se ir consultando as páginas de Cadastro de cada pessoa, ou se implementar um botão (próximo/anterior) na tela Cadastro da pessoa para se navegar para o próximo/anterior da lista com base no índice da *array* resultado. Caso se deseje estampar alguns dados resumo acerca da pessoa na tela resultado, será necessário mais um trabalho de busca na base de dados.

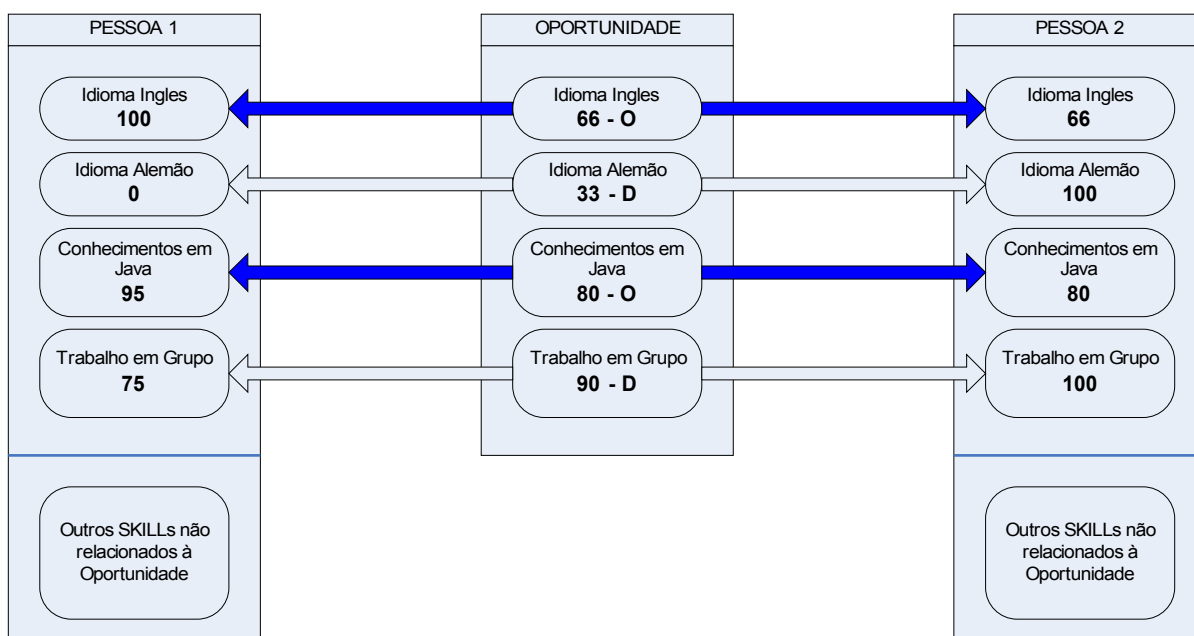


Figura 24 – Exemplo *Skill Score*

	Pessoa 1		Oportunidade	Pessoa 2	
	Skill Score	Skill Gap	Skill Score	Skill Score	Skill Gap
Obrigatórios	195	+33,6%	146	146	0,00%
Desejáveis	75	-39,0%	123	200	+62,6%
Total	270	+0,37%	269	346	+28,6%

Tabela 8 - Exemplo *Skill Score* e *Skill Gap*

A figura 24 e a tabela 8 mostram um exemplo de uma oportunidade e duas pessoas com os respectivos valores de *Skill Score* e de *Skill Gap*. No SACHA foi implementada a possibilidade de que o *ranking* seja baseado, como descrito anteriormente, no *Skill Score* obrigatório com desempate pelo *Skill Score* desejável, mas também a possibilidade de que o *ranking* se baseie na somatória obrigatório+desejável. No primeiro caso a Pessoa 1 estaria à frente da Pessoa 2 (195 a 146) e no segundo o oposto (270 a 346).

Em uma busca na base de dados por pessoas que atendam aos requisitos de uma dada oportunidade, tem-se duas opções de critério:

- 1 **Obrigatórios/Desejáveis:** compõe a lista de ranking de acordo com o *Skill Score* obrigatório, desempatando pelo *Skill Score* desejável.
- 2 **Obrigatórios+Desejáveis:** compõe a lista de ranking de acordo com o resultado da somatória dos *Skill Scores* obrigatórios e desejáveis.

150 pessoas
224/44

NOME	SCORE	SKILL GAP
PESSOA 4629	312 / 144	1,39 / 3,27
PESSOA 3134	312 / 133	1,39 / 3,02
PESSOA 3545	312 / 111	1,39 / 2,52
PESSOA 3051	312 / 99	1,39 / 2,25
PESSOA 1814	312 / 77	1,39 / 1,75
PESSOA 3491	312 / 44	1,39 / 1,00
PESSOA 1294	312 / 44	1,39 / 1,00
PESSOA 4739	301 / 166	1,34 / 3,77
PESSOA 3613	301 / 166	1,34 / 3,77
PESSOA 1697	301 / 144	1,34 / 3,27
PESSOA 300	301 / 133	1,34 / 3,02
PESSOA 4811	301 / 122	1,34 / 2,77
PESSOA 4782	301 / 111	1,34 / 2,52
PESSOA 3887	301 / 111	1,34 / 2,52
PESSOA 3199	301 / 111	1,34 / 2,52
PESSOA 4584	301 / 100	1,34 / 2,27
PESSOA 4766	301 / 89	1,34 / 2,02
PESSOA 4762	301 / 89	1,34 / 2,02
PESSOA 3406	301 / 89	1,34 / 2,02
PESSOA 2487	301 / 89	1,34 / 2,02
PESSOA 2112	301 / 89	1,34 / 2,02
PESSOA 4664	301 / 77	1,34 / 1,75
PESSOA 4430	301 / 77	1,34 / 1,75
PESSOA 2516	301 / 77	1,34 / 1,75
PESSOA 438	301 / 55	1,34 / 1,25
PESSOA 3911	301 / 55	1,34 / 1,25
PESSOA 1513	301 / 55	1,34 / 1,25
PESSOA 1462	301 / 55	1,34 / 1,25

Figura 25 – Lista de ranking obtida pelo critério 1

150 pessoas
224/44

NOME	SCORE	SKILL GAP
PESSOA 4739	467	2,08
PESSOA 3613	467	2,08
PESSOA 4629	456	2,04
PESSOA 3859	445	1,99
PESSOA 3134	445	1,99
PESSOA 1697	445	1,99
PESSOA 300	434	1,94
PESSOA 4811	423	1,89
PESSOA 4113	423	1,89
PESSOA 3545	423	1,89
PESSOA 2542	423	1,89
PESSOA 4782	412	1,84
PESSOA 4746	412	1,84
PESSOA 4582	412	1,84
PESSOA 3887	412	1,84
PESSOA 3481	412	1,84
PESSOA 3199	412	1,84
PESSOA 1739	412	1,84
PESSOA 1499	412	1,84
PESSOA 3051	411	1,83
PESSOA 4584	401	1,79
PESSOA 4564	401	1,79
PESSOA 4488	401	1,79
PESSOA 3093	401	1,79
PESSOA 2680	401	1,79
PESSOA 1689	401	1,79
PESSOA 712	390	1,74
PESSOA 4805	390	1,74

Figura 26 – Lista de ranking obtida pelo critério 2

Embora esta tela ainda não esteja com acabamento adequado no protótipo, pode-se ter uma idéia de como o modelo procede. No canto esquerdo superior tem-se a informação de quantas pessoas atenderam aos requisitos obrigatórios, e portanto entraram na lista de *ranking*, e também o *Skill Score* obrigatório e o desejável (150 pessoas; SS obrigatório = 224; SS desejável = 44). Observa-se que os primeiros colocados se alteraram nos dois critérios.

Observação: Os dados de teste são originários de base de dados real, portanto os nomes das pessoas foram omitidos.

Estes resultados podem sofrer mudanças de ordem pela manipulação do fator “Relevância”. Nos métodos de **skill.getScore()**, a rotina precisa implementar o seguinte cálculo:

$$SS_{final} = \frac{SS_{interno}}{10} \times relevância$$

Onde **SS**_{interno} é o calculo que deve ser feito internamente para que o *Skill Score* resulte normalizado na escala de 0 a 100. A relevância foi definida no protótipo como variando de 0 a 10 podendo chegar a 15, onde zero é um recurso para que um SKILL seja anulado na busca sem que haja a necessidade de exclusão, e 15 um recurso para se ampliar a relevância de um SKILL sem que haja a necessidade de se proceder ao rebaixamento da relevância de todos os demais SKILLS.

A Busca por Oportunidades representa o profissional ou aluno/estagiário desejando localizar na base de dados oportunidades às quais eles tenham possibilidades de concorrer.

Na tela de argumentos de busca por oportunidades é necessário definir quem é a pessoa, e existe uma opção de busca (Localizar Oportunidades) e uma opção de análise detalhada dos seus atributos em relação à determinada oportunidade (Verificar *Skill Gap*).

Na opção de localização de oportunidades não é feita uma análise que confronte SKILL pessoa com SKILL oportunidade. É calculado o *Skill Score* global obrigatório, tanto da oportunidade quanto o da pessoa e o sistema sinaliza na tela resultante se existe possibilidade ou não. Se o *Skill Score* total da pessoa é maior ou igual ao da oportunidade, então a possibilidade existe, mas como não há verificação SKILL a SKILL será necessário então a pessoa entrar na opção “Verificar *Skill Gap*” para que o detalhamento seja realizado.

Não foi implementado assim no SACHA, mas é tecnicamente possível que toda a confrontação SKILL a SKILL seja realizada para que a opção “Localizar Oportunidades” resulte exata, mas em termos práticos costuma-se evitar deixar disponível para uma população muito maior que a população de geradores de oportunidades opções de pesquisa que demandem muito custo de processamento.

OPORTUNIDADE	Score Opo	Score Pes	Skill Gap	
Estagiário TI	224	191	0,85	X
Genérico I	111	111	1,00	provável

Figura 27 - SACHA: Busca por Oportunidades

Na figura 27 o sistema apontou que de todas as oportunidades publicadas do nível “tecnólogo”, a “Estagiário TI” está fora das possibilidades (*Skill Score* obrigatório inferior ao da oportunidade), mas há possibilidades, pelo cômputo do *Skill Score* que a oportunidade “Genérico I” esteja dentro das possibilidades da pessoa.

SACHA: Busca por Oportunidades - Internet Explorer fornecido por Dell

http://localhost:17660/SACHA/busca_oportunidades_localizar.jsp

SACHA: Busca por Oportunidades

Sistema de Avaliação por Conhecimentos-Habilidades-Atitudes
*** Baseado em Gestão por Competências ***

Busca por Oportunidades
* Skill Gap *

[VOLTAR](#)

Estagiário TI

SKILL	ITEM	OBRIG (Oportun.)	DESEJ (Oportun.)	OBRIG (Pessoa)	DESEJ (Pessoa)	OBRIG (Skill Gap)	DESEJ (Skill Gap)	
Naturalidade	Data de Nascimento.	80	-	80	-	1,00	-	
Naturalidade	Local de Nascimento	-	-	-	-	-	-	
idioma - INGLES	Escreve	22	-	11	-	0,50	-	X
idioma - INGLES	Fala/Entende	22	-	11	-	0,50	-	X
idioma - INGLES	Lê	33	-	22	-	0,67	-	X
idioma - ESPANHOL	Escreve	-	-	-	-	-	-	
idioma - ESPANHOL	Fala/Entende	-	22	-	-	-	0,00	A
idioma - ESPANHOL	Lê	-	22	-	-	-	0,00	A
informatica usuario - EXCEL	Grau de Conhecimento >>	67	-	67	-	1,00	-	
informatica usuario - ACCESS	Grau de Conhecimento >>	-	67	-	33	-	0,49	A

Concluído

Internet | Modo Protegido: Ativado

16:16

Figura 28 - SACHA: Busca por Oportunidades

A figura 28 mostra em detalhes toda a contabilização SKILL a SKILL, item a item, apontando as reprovações e, nos itens desejáveis, mostrando onde está abaixo do “desejavelmente” esperado (A = atenção).

SACHA: Busca por Oportunidades - Internet Explorer fornecido por Dell

http://localhost:17660/SACHA/busca_oportunidades_localizar.jsp

SACHA: Busca por Oportunidades

Sistema de Avaliação por Conhecimentos-Habilidades-Atitudes
*** Baseado em Gestão por Competências ***

Busca por Oportunidades
* Skill Gap *

[VOLTAR](#)

Genérico I

S K I L L	I T E M	OBRIG (Oportun.)	DESEJ (Oportun.)	OBRIG (Pessoa)	DESEJ (Pessoa)	OBRIG (Skill Gap)	DESEJ (Skill Gap)
idioma - INGLES	Escreve	11	-	11	-	1,00	-
idioma - INGLES	Fala/Entende	11	-	11	-	1,00	-
idioma - INGLES	Lê	22	-	22	-	1,00	-
informatica usuario - EXCEL	Grau de Conhecimento >>	67	-	67	-	1,00	-

Concluído

Internet | Modo Protegido: Ativado

16:14

Figura 29 - SACHA: Busca por Oportunidades

A figura 29 mostra em detalhes a análise da oportunidade “Genérico I” em que há igualdade em todos os itens. Esta pessoa entraria para a lista de *ranking* e ficaria apenas na dependência de candidatos com *Skill Gap* mais favorável.

Ainda não foi implementado no SACHA, mas é interessante que haja mais um item de informação que seria justamente em que posição do *ranking* a pessoa estaria e em quais itens ela precisaria investir maiores esforços para aumentar sua competitividade no mercado de trabalho. Esta seria uma informação de cunho social de relevada importância em uma implementação real deste tipo de sistema, pois sinalizaria com mais precisão as reais condições atuais da pessoa e a ajudaria a decidir se investir em qualificação para aumentar a sua competitividade ou concentrar a procura em funções mais adequadas à sua condição atual.

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Este estudo levantou de forma consolidada o modelo da Competência (humana) segundo a teoria administrativa da Gestão por Competências, cuja aplicação nas empresas brasileiras, americanas e européias, nos dias atuais, está em franca expansão, idealizou e implementou um modelo computacional cujo objetivo foi representar o mais fielmente possível o modelo conceitual e servir como ferramenta de apoio à decisão em processos de avaliação de pessoas por Competência.

Embora o padrão HR-XML (2007) não especifique como deve ser modelada a implementação, procurou-se manter o modelo de persistência relacional dentro das recomendações deste padrão pelo seu forte enfoque empresarial de recursos humanos, e também não deixou de observar a estrutura proposta por Locatelli (2007) para representação e ponderação dos conceitos acadêmicos por Níveis de Aquisição do Conhecimento, embora este estudo não tenha abarcado a avaliação da Competência Acadêmica, mas deixando aberta a possibilidade para tal.

O protótipo implementado, batizado por SACHA (Sistema de Avaliação por Conhecimentos, Habilidades e Atitudes), mostrou-se funcional em todos os casos de uso propostos e demonstrou flexibilidade para acomodar diversos modelos de SKILLS, inclusive modelos parametrizáveis via arquivos externos. A inclusão de novos SKILLS não altera o código básico do sistema, apenas a classe de catalogação de SKILLS, o registro na base de dados e, se não houver classe já modelada para uso do novo SKILL, a instalação da nova classe no diretório correspondente.

A introdução dos conceitos de *Skill Score* e *Skill Gap* (AKEN, 2007), combinados com os fatores de obrigatoriedade e relevância, imprimiram grande flexibilidade para ajustes finos na obtenção de listas de *ranking* das pessoas mais adequadas (segundo o modelo) para ocuparem determinadas oportunidades oferecidas pelo mercado de trabalho, o que dá ao SACHA boas características para ganhar espaço na área dos sistemas adaptativos.

Embora tenha-se tomado por base estudos anteriores, o modelo a que se chegou neste estudo não representa uma continuidade e sim uma similaridade de contexto. De forma que os resultados são frutos de experimentos embrionários e

certamente carecem de aperfeiçoamentos além de complementos diversos conforme relacionados abaixo:

- Desenvolver o lado da Competência Acadêmica com base no modelo proposto por Locatelli (2007).
- Desenvolver o modelo acadêmico-profissional com vistas a estabelecer o elo escola-empresa.
- Validar e, se for o caso, aprimorar os mecanismos SQL para operação em grandes massas de dados.
- Ampliar a pesquisa na modelagem das classes de SKILL quanto aos tipos existentes na vida real.
- Ampliar a pesquisa na modelagem das classes de SKILL parametrizáveis de forma a aumentar a padronização e reduzir a necessidade de manutenção da codificação à medida da inclusão de um novo SKILL.
- Desenvolver estudo voltado à mineração de dados para o modelo proposto.
- Desenvolver a interface para permitir a comunicação com testes on-line, de forma a automatizar a atribuição da “nota” do SKILL para este tipo de caso.
- Desenvolver *driver* para troca de dados entre sistemas similares via XML segundo a padronização HR-XML.
- Analisar alternativas de projeto de software para tentar eliminar o problema da constante preocupação que o modelo possibilita quanto à ocorrência da “explosão de classes”.
- Desenvolver módulos com mais enfoque em Inteligência Artificial para os procedimentos de avaliação. Estudar a possibilidade de inserir parâmetros que possibilitem um determinado grau de subjetividade na avaliação (p. ex. pessoa sem um determinado requisito obrigatório mas que apresenta potencial para se desenvolver).
- Aprimorar a interface gráfica do SACHA, visando melhor interação Homem-Máquina.

REFERÊNCIAS

AKEN, Andrew; Michael D. Michalisin – *The Impact of the Skills Gap on the Recruitment of MIS Graduates* – St. Louis, Missouri USA: SIGMIS-CPR 19-21 abril 2007

BAILEY, Janet L.; Greg Stefaniak – *Industry Perceptios of the Knowledge, Skills, and Abilities Needed by Computer Programmers* – San Diego, CA USA: SIGCPR 2001.

CLARK, Thomas George – *Defining a Competency Framework to Shape the Professional Education of National Security Master Strategists: A Web-Based Delphi Study* – Dissertation to requirements for the degree of DOCTOR OF PHILOSOPHY - 2005

DUARTE, Fausto Afonso - *O Processo de Seleção de Pessoas nas Organizações Baseado no Conceito de Competência* – Dissertação de Mestrado – Pontifícia Universidade Católica – São Paulo - 2001

DUTRA, Joel Souza; André Luiz Fisher; José Antônio Monteiro Hipólito; Maria Tereza Leme Fleury; Marisa Eboli – *Gestão por Competências* – São Paulo: Editora Gente – 2001.

FLEURY, Afonso; Maria Tereza Leme Fleury – *Estratégias Empresariais e Formação de Competências – Um Quebra-cabeça caleidoscópico da Indústria Brasileira* – São Paulo: Editora Atlas – 2000.

HAMMER, Michael; James Champy – *Reengenharia – revolucionando a empresa em função dos clientes, da concorrência e das grandes mudanças da gerência* – Rio de Janeiro: Editora Campus – 1994.

HRXML Specification (2007) – disponível em http://ns.hr-xml.org/2_0/HR-XML-2_0/CPO/Competencies.pdf - acesso em 04/03/2008

IEEE Competency Working Group (2005) - disponível em <http://itsc.ieee.org/wg20/index.html> - acesso em 04/03/2008

IMS RDCEO (2002) - *IMS Reusable Definition of Competency or Educational Objective - Information Model Revision: 25 October 2002* – disponível em http://www.imsproject.org/competencies/rdceov1p0/imsrdceo_infov1p0.html - acesso em 04/03/2008

LOCATELLI, Mariana Brólio – *Ambiente de Apoio à Avaliação Continuada Baseado em Aprendizagem Significativa* – Dissertação de Mestrado – São Paulo: Universidade Presbiteriana Mackenzie – 2007.

NETO, Acácio Feliciano; José Davi Furlan; Wilson Higa – *Engenharia da Informação: Metodologia, Técnicas e Ferramentas* – São Paulo: Editora McGraw-Hill Ltda – 1988.

PARRY, SCOTTY B. (1996). *The quest for competencies: Competency studies can help you make HR decisions, but the results are only as good as the study* – disponível em http://www.trainingmag.com/msg/search/article_display.jsp?vnu_content_id=1505256&imw=Y – acesso em 29/03/2009

PERRENOUD, Philippe - *Novas competências para ensinar* - Porto Alegre: Editora Artmed – 2000.

PIMENTEL, Edison; OMAR, Nizam – *Um modelo para o Mapeamento do Conhecimento do Estudante em Ambientes Computacionais de Aprendizagem* – Tese de Doutorado – São José dos Campos: ITA – 2003.

RABAGLIO, Maria Odete - *Seleção por Competências* – São Paulo: Editora Educator – 2001

RABAGLIO, Maria Odete - *Ferramentas de Avaliação de Performance com Foco em Competências* – Rio de Janeiro: Editora Qualitymark – 2004.

SEABRA, Luis Antonio Moreira – *Simulação de Uma Linha de Montagem de Veículos: Uma aplicação em veículos militares* – Dissertação de mestrado – Rio de Janeiro: Pontifícia Universidade Católica - 2006

TSUNO, André Sussumi – *Um modelo de Sistemas Integrados para o Desenvolvimento e Gestão do Conhecimento Organizacional* – Dissertação de Mestrado – São Paulo: Universidade Presbiteriana Mackenzie –2007.

APÊNDICE A – Pesquisa preliminar na Internet e via e-mail para verificação aproximada da situação atual da utilização de Bancos de Competências no Brasil.

Para pesquisa documental via Internet foram utilizados os mecanismos de busca mais utilizados na atualidade, o Google (2008) e o Yahoo (2008). Foram submetidas à busca as expressões “banco de competências” e “banco de dados de competências”. Os resultados mais significativos, no entanto, vieram de sítios governamentais. Empresas da iniciativa privada aparentemente não divulgam este tipo de situação na Web.

	CATEGORIAS	%
A	Afirma ter necessidade/intenção de implantar um banco de competências	45
B	Afirma possuir um banco de competências interno	20
C	Aparenta ser um Banco de Currículos convencional com o rótulo de Banco de Competências	21
D	Banco de competências de acesso público integrando empresa-escola	2
E	Documento genérico comentando a importância de um banco de competências	4
F	Software comercial que diz ter um módulo de Banco de Competências	4
G	Simple cadastro de pessoas	4

Tabela 9 – Resultado da pesquisa na Internet quanto à existência de Portais de relacionamento baseados no modelo de Competências.

Nenhum dos resultados fornece informações técnicas e todos ressaltam apenas a palavra “Competência”.

Com relação à iniciativa privada, foram enviados e-mails aos responsáveis pelos departamentos de Recursos Humanos de empresas de médio e grande porte, nacionais e multinacionais, para se avaliar a situação em que está a adoção da “Gestão por Competências” e se há uma banco de dados baseado neste modelo.

11 foram as empresas respondentes:

BANCO ABN AMRO

Instituição bancária

<http://br.abnamro.com>

BANCO BRASCAN

Instituição bancária

<http://www.bancobrascan.com.br>

BEAUTY SERVICES

Organização de comercialização de perfumaria

<http://www.beautyservices.com.br>

COMPANHIA BRASILEIRA DE SOLUCOES E SERVICOS (VISA VALE)

Serviços de alimentação

<http://www.visavale.com.br>

DEUTSCHE BANK S.A.- BANCO ALEMAO

Instituição bancária

<http://www.db.com>

DIAGNÓSTICOS DA AMÉRICA

Conglomerado de Análises Laboratoriais

<http://diagnosticosdaamerica.com.br>

EDITORIA GLOBO S.A.

Empresa do segmento editorial

<http://editoraglobo.globo.com/>

GRUPO SCHINCARIOL

Fabricante de bebidas

<http://www.schincariol.com.br/>

GRUPO IBOPE

Instituição do segmento de pesquisas de opinião pública

<http://www.ibope.com.br>

GRUPO HERCULES

Empresa do ramo químico

<http://www.herc.com/>

VISA DO BRASIL EMPREENDIMENTOS LTDA

Instituição financeira do segmento de cartões de crédito

<http://visa.com>

Com relação à adoção da “Gestão por Competências” as respostas se dividiram em 3 grupos praticamente iguais:

1/3 – Não adota “Gestão por Competências.

1/3 – Está em processo de implantação.

1/3 – Já foi implantada.

Com relação à utilização de um sistema de banco de dados para gerenciar talentos, apenas 1/3 declararam ter algum tipo de recurso de banco de dados, algumas somente para pessoal interno, outras para captação de currículos de novos candidatos.

Com relação ao modelo do banco de dados ser orientado à competências 2/3 não responderam, 1/3 afirmou tratar-se de um banco de currículos convencional e apenas uma resposta afirmando tratar-se de modelo baseado no CHA, mas houve a possibilidade de se fazer esta verificação.

Conclusão: Percebe-se que, em termos tecnológicos, a questão encontra-se em estágio bastante inicial, provavelmente aguardando o estabelecimento da adoção administrativa para que os desenvolvimentos tecnológicos possam ser orientados.

APÊNDICE B – Projeto e Normalização do Modelo de Dados Relacional - Pessoas e SKILLS.

Para este procedimento vide Neto (1988).

Atributos necessários:

- **Pessoa:** atributo chave, nome, endereço, documentos etc.
- **SKILLS existentes no domínio:** atributo chave, nome, descrição, relação de itens.
- **SKILLS da Pessoa:** atributo identificador da Pessoa, atributo identificador do SKILL, valor.

As tabelas necessárias, em princípio, decorrem dos objetos existentes:

- PESSOAS
- SKILLS
- PESSOAS_SKILLS

Atributos da tabela PESSOA:

Identificador	Descrição	Tipo	Domínio
© NR	Número sequencial identificador da pessoa	NUMÉRICO INTEIRO	1 a + infinito
NM	Nome da pessoa	ALFANUMÉRICO	String
SX	Sexo da pessoa	CHARACTER	{“M”, “F”}
DT_NASC	Data de nascimento	DATA	01/01/1900 ao limite do SGBDR
DOC1 DOC2 ... DOCn	Sequência de atributos referentes à documentação.	ALFANUMÉRICO	String
END1 END2 ... ENDn	Sequência de atributos referentes ao endereço de residência.	ALFANUMÉRICO	String
OUTRO1 OUTRO2 ...	Outras informações acerca da pessoa que se fizerem necessárias	À definir	À definir

Tabela 10 – Normalização: Atributos da Tabela “pessoa”

Observação: o símbolo © indicará um atributo chave e o símbolo ® representará um atributo de chave estrangeira.

Primeira forma normal: Verificação de tabelas aninhadas (informações repetitivas que impliquem em repetição do atributo chave)

Há a possibilidade ou não de se considerar o conjunto de atributos “Documentos” ou o conjunto “Endereço” como um sub-grupo que estaria aninhado no grupo principal, podendo ser movido, cada um deles, para uma tabela à parte, de forma que a introdução de um novo documento, por exemplo, não implicasse em manutenção na estrutura da base de dados. Mas é sabido pela experiência prática (experiência do autor) que tabelas com grande quantidade de atributos em que há interesse de busca, acabam com grande quantidade de índices para tornar mais ágil a localização dos registros de interesse, mas o desempenho nas atualizações diminui, deixando a base de dados inflada com dados de índices, tornando as manutenções e procedimentos de backup cada vez mais trabalhosos.

Para estes casos é melhor criar uma tabela secundária com um atributo de relacionamento com a tabela pai, um atributo identificador dos atributos movidos e um atributo contendo seu valor. Para tanto pode-se criar as tabelas PESSOA_DOCS e PESSOAS_END

Atributos da tabela PESSOA_DOCS

Identificador	Descrição	Tipo	Domínio
© ® PESSOA_NR	Atributo chave herdado da tabela-pai (NR)	Vide tabela PESSOA	
© CD	Código identificador do documento	ALFANUMÉRICO	String
VL	Valor do atributo (no caso o número do documento)	ALFANUMÉRICO	String

Tabela 11 – Normalização: Atributos da Tabela “pessoa_docs”

O atributo identificador do documento é o CD, e o que o relaciona com determinada pessoa é o PESSOA_NR, portanto temos uma chave concatenada PESSOA_NR + CD. Há SGBDRs que criam índices múltiplos a partir do índice da chave primária, permitindo a pesquisa por um único campo de uma chave concatenada, mas outros não tem esta implementação, exigindo que, além do índice da chave primária, se crie um índice separado para cada atributo da chave que se tenha interesse de busca. Mas de qualquer forma não haverá uma grande quantidade de índices a serem mantidos pelo SGBDR, já que o número independe do número de documentos existentes.

A tabela PESSOA_END poderia ter estrutura idêntica, sendo que no atributo CD, ao invés de se ter o conteúdo “CPF” se teria, por exemplo, “LOGRADOURO”.

Neste conceito se poderia até criar um única tabela, PESSOA_DADOS, com esta estrutura, contendo todas as informações complementares.

Segunda forma normal: eliminação de dependências parciais em chaves concatenadas.

A tabela PESSOA não tem chave concatenada, por conseguinte já se encontra na segunda forma normal. Na prática procura-se eleger um atributo (ou grupo de atributos) para ser o chave de forma a identificar, de fato, o objeto em questão. Para esta missão entram em campo os especialistas de domínio que irão definir que atributos melhor identificam o objeto. Para o caso de pessoas, no caso de maiores de idade, costuma-se utilizar o documento CPF, mas esta questão está fora do escopo deste estudo e, para não criar dificuldades que não tem a ver com o problema, utilizaremos o atributo numérico sequencial.

Terceira forma normal: eliminação de dependências transitivas.

São atributos resultados de cálculos realizados com outros existentes, podendo ser eliminados, ou dependentes de atributos de chaves estrangeiras, devendo ser movidos para a tabela respectiva. Não há este caso, estando a tabela já na terceira forma normal.

Atributos da tabela SKILL

Identificador	Descrição	Tipo	Domínio
© NR	Número sequencial identificador do SKILL	NUMÉRICO INTEIRO	1 a + infinito
NM	Nome ou descrição do SKILL	ALFANUMÉRICO	String
CD_ITEM	Código do item do SKILL	ALFANUMÉRICO	String
DS_ITEM	Nome ou descrição do item, podendo inclusive citar o domínio de valores.	ALFANUMÉRICO	String

Tabela 12 – Normalização: Atributos da Tabela “skill”

Primeira forma normal: Neste caso o SKILL pode ter mais de um item (CD_ITEM), de forma que haverá repetição do atributo chave e do nome do SKILL para cada item. As informações referentes aos itens deverão ser movidas para uma nova tabela → SKILL_ITEM.

Atributos da nova tabela SKILL_ITEM

Identificador	Descrição	Tipo	Domínio
© ® SKILL_NR	Atributo chave herdado da tabela-pai (NR)	Vide tabela SKILL	
© CD	Código do item do SKILL	ALFANUMÉRICO	String
DS	Nome ou descrição do item, podendo inclusive citar o domínio de valores.	ALFANUMÉRICO	String

Tabela 13 – Normalização: Atributos da Tabela “skill_item”

Primeira forma normal: só existe um atributo fora da chave, logo não haverá repetições → já está na primeira forma normal.

Segunda forma normal: não há dependências parciais da chave concatenada, pois o único atributo (DS) depende de ambos os atributos da chave.

Terceira forma normal: Não há dependências transitivas. O campo DS é atômico.

Voltando à tabela SKILL:

Segunda forma normal: não há chave concatenada → já está na segunda forma normal.

Terceira forma normal: Não há dependências transitivas. O campo NM é atômico.

Os SKILLS pessoais devem estar reunidos em uma tabela chamada PESSOA_SKILL. A tabela SKILL, no entanto, não contém a relação dos itens componentes de um dado SKILL, portanto o relacionamento deverá ser feito com a tabela SKILL_ITEM.

Atributos da tabela PESSOA_SKILL_ITEM:

Identificador	Descrição	Tipo	Domínio
© ® PESSOA_NR	Atributo chave herdado da tabela-pai (NR)	Vide tabela PESSOA	
© ® SKILL_NR	Atributo chave herdado da tabela-pai (NR)	Vide tabela SKILL_ITEM	
© ® SKILL_ITEM_CD	Atributo chave herdado da tabela-pai (CD)	Vide tabela SKILL_ITEM	
SKILL_ITEM_VLN	Atributo com o valor do item (caso numérico)	NUMÉRICO	- infinito a + infinito
SKILL_ITEM_VLA	Atributo com o valor do item (caso alfanumérico)	ALFANUMÉRICOI	String

Tabela 14 – Normalização: Atributos da Tabela “*peessoa_skill_item*”

Para completar o modelo só falta a tabela para armazenar o valor resultante (quando houver) dos itens do SKILL. Abaixo temos o modelo físico, com os ajustes práticos necessários, gerado a partir do software freeware DB Designer 4 versão 4.0.5.6 para o SGBDR MySQL.

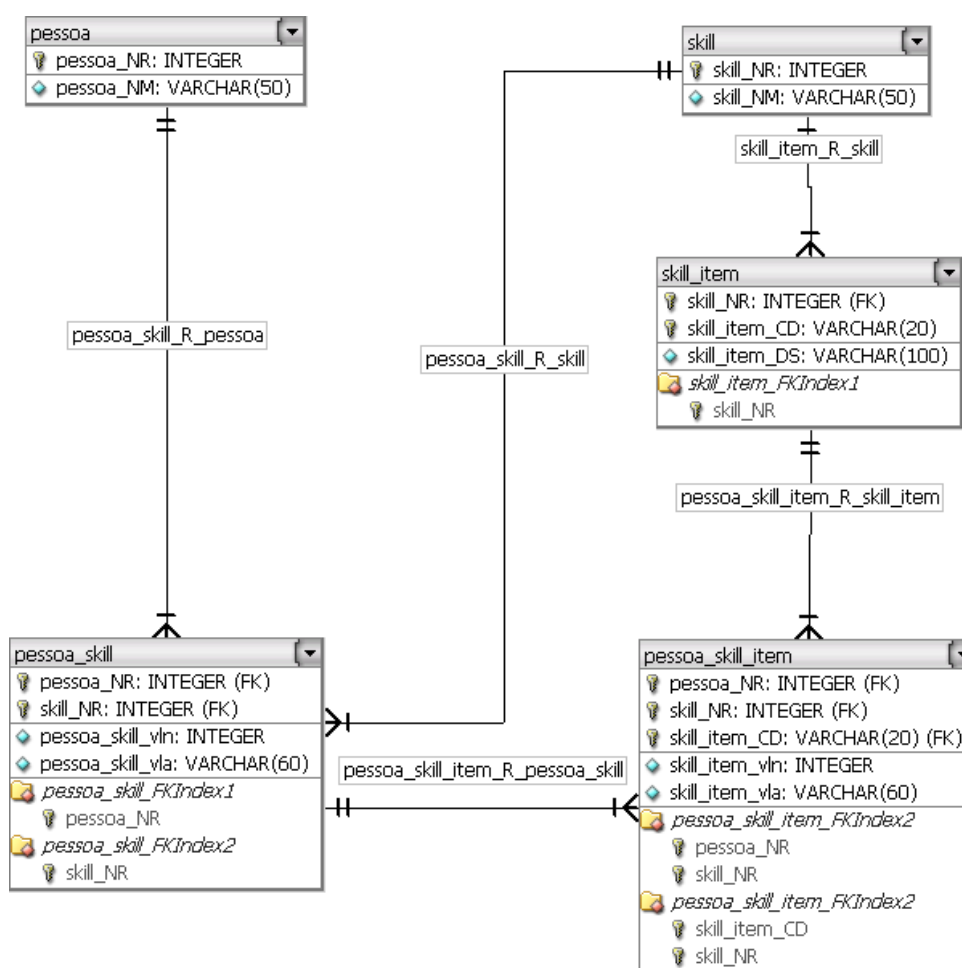


Figura 30 – DER Pessoa-Skill

Script de criação da base:

```
CREATE TABLE skill (
  skill_NR INTEGER UNSIGNED NOT NULL,
  skill_NM VARCHAR(50) NULL,
  PRIMARY KEY(skill_NR)
)
TYPE=InnoDB;
```

```

CREATE TABLE pessoa (
  pessoa_NR INTEGER UNSIGNED NOT NULL,
  pessoa_NM VARCHAR(50) NULL,
  PRIMARY KEY(pessoa_NR)
)
TYPE=InnoDB;

CREATE TABLE skill_item (
  skill_NR INTEGER UNSIGNED NOT NULL,
  skill_item_CD VARCHAR(20) NOT NULL,
  skill_item_DS VARCHAR(100) NULL,
  PRIMARY KEY(skill_NR, skill_item_CD),
  INDEX skill_item_FKIndex1(skill_NR),
  FOREIGN KEY(skill_NR)
  REFERENCES skill(skill_NR)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
)
TYPE=InnoDB;

CREATE TABLE pessoa_skill (
  pessoa_NR INTEGER UNSIGNED NOT NULL,
  skill_NR INTEGER UNSIGNED NOT NULL,
  pessoa_skill_vln INTEGER NOT NULL,

  pessoa_skill_vla VARCHAR(60) NULL,
  PRIMARY KEY(pessoa_NR, skill_NR),
  INDEX pessoa_skill_FKIndex1(pessoa_NR),
  INDEX pessoa_skill_FKIndex2(skill_NR),
  FOREIGN KEY(pessoa_NR)
  REFERENCES pessoa(pessoa_NR)
  ON DELETE RESTRICT
  ON UPDATE CASCADE,
  FOREIGN KEY(skill_NR)
  REFERENCES skill(skill_NR)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
)
TYPE=InnoDB;

CREATE TABLE pessoa_skill_item (
  pessoa_NR INTEGER UNSIGNED NOT NULL,
  skill_NR INTEGER UNSIGNED NOT NULL,

  skill_item_CD VARCHAR(20) NOT NULL,
  skill_item_vln INTEGER UNSIGNED NULL,
  skill_item_vla VARCHAR(60) NULL,
  PRIMARY KEY(pessoa_NR, skill_NR, skill_item_CD),
  INDEX pessoa_skill_item_FKIndex1(pessoa_NR, skill_NR),
  INDEX pessoa_skill_item_FKIndex2(skill_NR, skill_item_CD),
  FOREIGN KEY(pessoa_NR, skill_NR)
  REFERENCES pessoa_skill(pessoa_NR, skill_NR)
  ON DELETE RESTRICT
  ON UPDATE CASCADE,
  FOREIGN KEY(skill_NR, skill_item_CD)
  REFERENCES skill_item(skill_NR, skill_item_CD)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
)
TYPE=InnoDB;

```

Modificações posteriores:

1. Incluída coluna para identificar se o SKILL se trata de um Conhecimento, Habilidade ou Atitude. Na tabela “skill” → coluna “skill_CHA”, char(1); domínio = {“C”, “H”, “A”}.
2. Incluída a tabela “skill_item_frequencia” que armazenará a distribuição em frequência dos SKILLS na base de dados, com objetivo de auxiliar a lógica de busca a iniciar pelos SKILLS obrigatórios de menor incidência.

```
CREATE TABLE skill_item_frequencia (
  skill_NR INTEGER UNSIGNED NOT NULL,
  freq INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(skill_NR),
  INDEX skill_frequencia_FKIndex1(skill_NR),
  INDEX skill_item_frequencia_index1(freq),
  FOREIGN KEY(skill_NR)
    REFERENCES skill(skill_NR)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
)
TYPE=InnoDB;
```

3. Incluídas as tabelas “logica” e “skill_logica” que relacionarão um determinado SKILL a uma determinada lógica de busca.

```
CREATE TABLE logica (
  logica_NR INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  logica_CD VARCHAR(20) NULL,
  logica_DS TEXT NULL,
  PRIMARY KEY(logica_NR),
  UNIQUE INDEX logica_index1(logica_CD)
)
TYPE=InnoDB;

CREATE TABLE skill_logica (
  skill_NR INTEGER UNSIGNED NOT NULL,
  logica_NR INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(skill_NR),
  INDEX skill_logica_FKIndex1(skill_NR),
  INDEX skill_logica_FKIndex2(logica_NR),
  FOREIGN KEY(skill_NR)
    REFERENCES skill(skill_NR)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  FOREIGN KEY(logica_NR)
    REFERENCES logica(logica_NR)
    ON DELETE NO ACTION
    ON UPDATE CASCADE
)
TYPE=InnoDB;
```

O modelo da Oportunidade segue o mesmo raciocínio acrescido de alguns campos que não interferem na normalização.

ANEXO A – Padrão HR-XML – Exemplos

(Trata-se do APENDICE C da especificação HR-XML 2007).

7 Appendix C – Reference Examples

7.1 Competency with Years of Experience and Test Score

This example shows an employee who has Java as a competency. Acme Company, using their standard Java test, tested this employee. The candidate's test score was 89 on that test. In addition, it's recorded that the employee has 4 years of experience using Java and on a scale from 1-100, has a score of 90 for their level of interest.

```
<?xml version="1.0" encoding="UTF-8"?>
<Competency xmlns="http://ns.hr-xml.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ns.hr-xml.org Competencies.xsd" description="Java is an object oriented computer
  language" name="Java">

  <CompetencyId description="Competency id is based on Acme internal taxonomy" id="574"
    idOwner="Acme Company"/>

  <TaxonomyId description="My ids are based on Acme Company Taxonomy" idOwner="Acme Company"
    id="1"/>

  <CompetencyEvidence dateOfIncident="2001-08-23" name="Test Score" typeDescription="Test Score
    from internal test" typeId="54">
    <EvidenceId description="Java Test from internally administered test" id="547" idOwner="Acme
      Company"/>
    <NumericValue description="100 point scale" maxValue="100"
      minValue="0">89</NumericValue>
  </CompetencyEvidence>

  <CompetencyEvidence dateOfIncident="2001-08-23" name="Years of Experience"
    typeDescription="Years of Experience" typeId="7">
    <EvidenceId description="Years of Experience in Competency" id="7" idOwner="Acme
      Company"/>
    <NumericValue description="Range in years for experience">4</NumericValue>
  </CompetencyEvidence>

  <CompetencyWeight type="levelOfInterest">
    <NumericValue description="Acme Company Scale 100 point" maxValue="100"
      minValue="0">90</NumericValue>
  </CompetencyWeight>
</Competency>
```

(...)

7.4 Recursive Competencies: Communication Skills

Communication skills can be made up of two skills: written and oral communication skills. Written and oral communication skills are the measurable and observable skills to which an employee/applicant will be measured. In order to assess whether a person has Communication Skills, it is necessary to evaluate the person's written and oral communication skills (the measurable and observable skills). This example weights oral communication skills higher (65 percent) than written communication skills (35 percent).

```
<?xml version="1.0"?>
```

```
<Competency name="Communication Skills" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://ns.hrxml.org" xsi:schemaLocation="http://ns.hr-xml.org
Competencies.xsd">
```

```
  <Competency name="Written Communication Skills">
    <CompetencyEvidence name="WRITTENTEST1-A" dateOfIncident="1995-01-01"
lastUsed="2000-01-01">
      <NumericValue minValue="3" maxValue="5" description="SEP-equivalent Skill-Level
Range">5</NumericValue>
    </CompetencyEvidence>
    <CompetencyWeight>
      <NumericValue minValue="0" maxValue="100">35</NumericValue>
    </CompetencyWeight>
  </Competency>
```

```
  <Competency name="Oral Communication Skills">
    <CompetencyEvidence name="ManagerObservation" dateOfIncident="1996-01-01"
lastUsed="2000-01-01">
      <NumericValue minValue="1" maxValue="5" description="Company XYZ Skill
Range">5</NumericValue>
    </CompetencyEvidence>
    <CompetencyWeight>
      <NumericValue minValue="0" maxValue="100">65</NumericValue>
    </CompetencyWeight>
  </Competency>
```

```
</Competency>
```

```
(...)
```