



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS PONTA GROSSA
DEPARTAMENTO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO
PPGEP

DANI JULIANO CZELUSNIAK

PROPOSTA DE SISTEMA PARA APOIO À GESTÃO
DE COMPETÊNCIAS BASEADA EM SISTEMAS
AGENTES

PONTA GROSSA
DEZEMBRO - 2007

Livros Grátis

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

Milhares de livros grátis para download.

DANI JULIANO CZELUSNIAK

**PROPOSTA DE SISTEMA PARA APOIO À GESTÃO
DE COMPETÊNCIAS BASEADA EM SISTEMAS
AGENTES**

Dissertação apresentada como requisito parcial à obtenção do título de Mestre em Engenharia de Produção, do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração: Gestão Industrial, do Departamento de Pesquisa e Pós-Graduação, do Campus Ponta Grossa, da UTFPR.

Orientador: Prof. Dario Eduardo Amaral Dergint, Dr.

PONTA GROSSA

DEZEMBRO - 2007

C998 Czelusniak, Dani Juliano

Proposta de sistema para apoio à gestão de competências baseada em sistemas agentes. / Dani Juliano Czelusniak. -- Ponta Grossa: [s.n.], 2007.

126 f.: il. ; 30 cm.

Orientador: Prof. Dr. Dario Eduardo Amaral Dergint

Dissertação (Mestrado em Engenharia da Produção) - Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa. Programa de Pós-Graduação em Engenharia de Produção. Ponta Grossa, 2007.

1. Gestão de competências. 2. Desempenho. 3. Sistemas agentes. 4. Frameworks. I. Dergint, Dario Eduardo Amaral. II. Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa. III. Título.

CDD 658.3125

Dedico este trabalho à todos aqueles que acreditaram e conseguiram. Em especial minha família e minha esposa.

“Se vi mais longe, foi porque estava
sobre ombros de gigantes.”

Carta de Isaac Newton à Robert Hooke. 1676.

RESUMO

Observando-se as novas tecnologias para desenvolvimento de software presentes no mercado, uma opção em especial vêm ganhando destaque pela possibilidade de se construir sistemas informatizados com módulos que trabalham de forma autônoma. Estes softwares são chamados de sistemas agentes. Neste sentido, o objetivo deste trabalho é verificar a viabilidade do desenvolvimento e ensaio de um protótipo de sistema informatizado para auxílio ao processo de gestão de competências em cursos de Pós-Graduação em Engenharia de Produção. Este sistema será norteado pela tecnologia agentes através da utilização do *framework* JADE com a linguagem de programação C# .NET para a leitura de currículos XML exportados pelo sistema de Currículos Lattes. Neste processo, foram utilizadas estas tecnologias com a finalidade de sedimentar uma base tecnológica inovadora para a criação de uma ferramenta informatizada que permita extrair informações, com o intuito de auxiliar o gestor durante o processo de busca e análise de indicadores de competências. As metodologias da pesquisa utilizadas são a revisão bibliográfica, na conceituação das teorias que regem a gestão de competências e os sistemas agentes e, estado da arte, na proposta, desenvolvimento e ensaio da ferramenta informatizada. Aduz-se que o uso da metodologia baseada em agentes pode proporcionar resultados satisfatórios por ser flexível e oferecer formas de tratar características humanas de forma inovadora, criando soluções modulares que têm a capacidade de tratar grandes volumes de dados de forma associativa.

Palavras-chave: Gestão de Competências, Sistemas Agentes, Tecnologia, JADE.

ABSTRACT

Observing the new technologies for software development, an special option that comes gaining prominence, for building software with modules that works in autonomous way. This particular kind of software is called "Agent Systems". By this way, the goal of this work is to verify the viability of development and tests of a software archetype to aid the competence management process in Post-Graduate of Production Engeneering Courses. This system will be designed using JADE Agent Framework, using the development language C# .NET, to read the XML files exported by the Curriculo Lattes system. In this process, the technologies have been use to build a innovative technology tool that aims the manager to extract some kinds of information, during the seek and analysis process of competence's indicators. The research methodology used is the bibliographical revision for appraise the theories as competence management and agent systems and state of art when making the proposal, development and tests of the software. This work expose that the agent based methodology can provides satisfactory results for being flexible and to offer forms to deal with human characteristics, beings in a innovative form, creating modular solutions that have the capacity to deal with great databases in a associative form.

Keywords: Competence Management, Agents Systems, Jade Technology.

LISTA DE FIGURAS

Figura 1 – Composição da Engenharia de Produção.....	1
Figura 2 – Convergência das diversas áreas para a Engenharia de Produção	2
Figura 3 – Formação da competência sob a ótica americana. Fonte: Durand (1998)	12
Figura 4 – Formação da competência sob a ótica francesa. Fonte: Durand (1998)	12
Figura 5 – Seleção de Pessoal. Fonte: Chiavenato (2004, p. 132).....	20
Figura 6 – Competências SCoM. Fonte: (SIEMENS, 2006, p.04).....	22
Figura 7 – Elementos formadores da competência SCoM. Fonte: (SIEMENS, 2006, p.05) .	22
Figura 8 – Habilidades SCoM. Fonte: (SIEMENS, 2006, p.06).....	23
Figura 9 – Composição das competências SCoM. Fonte: (SIEMENS, 2006, p.07).....	24
Figura 10 – Estratégia organizacional SCoM. Fonte: (SIEMENS, 2006, p.08)	24
Figura 11 – Processo SCoM. Fonte (SIEMENS, 2006, p.9-19).....	25
Figura 12 – Avaliação 360 graus. Fonte: Efix Avaliações (2005, p. 07).....	27
Figura 13 – Exemplo de Pseudocódigo.....	31
Figura 14 – Exemplo de Fluxograma.....	32
Figura 15 – Arquivo XML visualizado em modo "Código Fonte"	40
Figura 16 – Arquivo XML visualizado em modo "Blocos de Dados".....	41
Figura 17 – Questionário de Identificação de Competências. Fonte: Portal Efix	49
Figura 18 – Avaliação por Competências – Perfil do Colaborador. Fonte: Portal Efix	50
Figura 19 – Quantificação da Competência - Índícios. Fonte: Portal Efix	51
Figura 20 – Avaliação de Competências - Resultado. Fonte: Portal Efix	52
Figura 21 – Competências no Sistema Athena	53
Figura 22 – Sistema Athena	54
Figura 23 – Habilidades SCoM.....	55
Figura 24 – Cadastro de dados no Athena.....	56
Figura 25 – Composição dos Sistemas Lattes	58
Figura 26 – Currículo Lattes - Cadastro inicial	59
Figura 27 – Tela inicial do Currículo Lattes	60
Figura 28 – Formação Acadêmica e Titulação	61
Figura 29 – Exportação de currículo em formato XML	62
Figura 30 – Envio de Currículo para o IS Curriculum (Fonte: MARCHEZAN, 2006).....	62
Figura 31 – Recepção de Currículo Lattes no IS Curriculum (Fonte: MARCHEZAN, 2006).	63
Figura 32 – Serviços do Portal IS Curriculum (Fonte: MARCHEZAN, 2006)	64
Figura 33 – Visualizador de Currículos (Fonte: MARCHEZAN, 2006)	65

Figura 34 – Demografia Curricular (Fonte: MARCHEZAN, 2006)	66
Figura 35 – Classificador Curricular (Fonte: MARCHEZAN, 2006)	67
Figura 36 – Portal Inovação.....	68
Figura 37 – Busca por competências no Portal Inovação	69
Figura 38 – Canal PLSQL do Currículo Lattes	70
Figura 39 – Javascript no Currículo Lattes	71
Figura 40 – Exemplo de sistema agentes	77
Figura 41 – Estrutura Interna de um Agente (adaptado de AOKI, 1999 p. 34)	78
Figura 42 – Modelo estrutural básico do JADE (BELLIFEMINE, 2007, p.07)	81
Figura 43 – Plataforma Agente e containers (CAIRE, 2003, p. 05).....	82
Figura 44 – Ciclo de vida de um agente (Bellifemine, 2007, p. 11).....	83
Figura 45 – Diagrama de Casos de Uso	91
Figura 46 – Diagrama de Seqüência	92
Figura 47 – JADE com Visual Studio Express.....	94
Figura 48 – Estrutura de execução do agente JADE (CAIRE, 2003, p.10).....	96
Figura 49 – Troca de mensagens entre agentes (CAIRE, 2003, p.14).	98
Figura 50 – Ativação dos agentes	100
Figura 51 – Diagrama do Sistema Hydra	101
Figura 52 – Armazenamento das buscas no componente	109
Figura 53 – Ensaio com o termo chave "Qualidade"	110
Figura 54 – Ensaio com o termo chave "Manutenção"	111
Figura 55 – Ensaio com o termo chave "Produção"	112
Figura 56 – Ensaio com o termo chave "Produto"	113

LISTA DE QUADROS

Quadro 1 – Linguagens de programação do paradigma estruturado.....	33
Quadro 2 – Linguagens de programação do paradigma orientado à objetos	36
Quadro 3 – Bancos de dados e suas características	42
Quadro 4 – Relação sistemas vs. dificuldades apresentadas	74

LISTA DE SIGLAS

AMS	<i>Agent Management System</i>
ASP	<i>Active Server Pages</i>
BCPL	<i>Basic Combined Programming Language</i>
CAC	<i>Agent Communication Channel</i>
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DF	<i>Directory Facilitator</i>
DTD	<i>Document Type Definition</i>
EQMap	<i>Emotional Quotient Map</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
FIPA-SL0	Linguagem padrão de comunicação entre agentes JADE
HTML	<i>Hypertext Markup Language</i>
ICTIs	Instituições de Ciência, Tecnologia e Inovação
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IL	Intermediate Language
I&E&P	Instituição de Ensino e Pesquisa
JADE	<i>Java Agent Development Environment</i>
OOP	<i>Object Oriented Programming</i>
PL/SQL	<i>Procedural Language / Structured Query Language</i>
PL/PGSQL	<i>Procedural Language / PostgreSQL Structured Query Language</i>
PPGEP	Programa de Pós-Graduação em Engenharia de Produção
SCoM	<i>Score Competence Management</i>
SGBD	Sistema Gerenciador de Banco de Dados
SGML	<i>Standard Generalized Markup Language</i>
SQL	<i>Structured Query Language</i>
UFPR	Universidade Federal do Paraná
UTFPR	Universidade Tecnológica Federal do Paraná
VM	<i>Virtual Machine</i>
XML	<i>Extended Markup Language</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

RESUMO.....	v
ABSTRACT.....	vi
LISTA DE FIGURAS.....	vii
LISTA DE QUADROS.....	ix
LISTA DE SIGLAS.....	x
1 INTRODUÇÃO	1
1.1 APRESENTAÇÃO	1
1.2 JUSTIFICATIVA E MOTIVAÇÃO	2
1.3 OBJETIVOS.....	3
1.4 METODOLOGIA DE PESQUISA	4
1.5 ESTRUTURA DO TRABALHO	5
2 REVISÃO BIBLIOGRÁFICA.....	7
2.1 CONCEITO DE COMPETÊNCIA	7
2.1.1 Bases históricas e evolução do termo competência.....	9
2.1.2 Competências e a Teoria da Complexidade.....	14
2.1.3 Visão da Teoria da Complexidade na busca por indicadores de competências	17
2.1.4 Metodologias para busca de indicadores de competências	19
2.1.5 Considerações sobre competências.....	27
2.2 LINGUAGENS DE PROGRAMAÇÃO E BASES DE DADOS UTILIZADOS NA CONSTRUÇÃO DE SISTEMAS	29
2.2.1 Conceitos de linguagens de programação	30
2.2.2 Conceitos de bases de dados	36
2.2.3 Considerações sobre linguagens de programação e bases de dados.....	42
2.3 SISTEMAS INFORMATIZADOS PARA AUXÍLIO À GESTÃO DE COMPETÊNCIAS 44	
2.3.1 Sistema Efix Competências.....	47
2.3.2 Sistema Athena	52
2.3.3 Sistemas Informatizados do Instituto Stela.....	57
2.3.4 Considerações sobre os sistemas existentes de auxílio à Gestão de Competências... 71	
2.4 TECNOLOGIA AGENTES NA CONSTRUÇÃO DE SOFTWARE	75
2.4.1 Conceitos de sistemas agentes.....	76
2.4.2 Framework JADE – <i>Java Agent Development Environment</i>	79
2.4.3 Considerações sobre sistemas agentes.....	85
2.5 CONSIDERAÇÕES SOBRE O REFERENCIAL TEÓRICO.....	86
3 PROPOSTA DE SISTEMA AGENTES PARA BUSCA DE INDÍCIOS DE COMPETÊNCIAS DE EGRESSOS DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO	89

3.1	TECNOLOGIAS E METODOLOGIAS ADOTADAS	89
3.2	<i>REQUISITOS DO SISTEMA</i>	91
3.3	CODIFICAÇÃO DO PROTÓTIPO EM LINGUAGEM C# COM JADE <i>LEAP</i>	93
3.3.1	Agentes JADE em linguagem C#	94
3.3.2	Estrutura do protótipo	99
	CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO	105
4	ENSAIO DO PROTÓTIPO AGENTES JUNTO AO PPGE	108
4.1	O PROCESSO DE ENSAIO	108
4.2	CONSIDERAÇÕES SOBRE O ENSAIO DO SISTEMA AGENTES	113
5	CONSIDERAÇÕES FINAIS	116
6	REFERÊNCIAS BIBLIOGRÁFICAS	120

1 INTRODUÇÃO

Nesta seção se apresenta a justificativa, objetivos e metodologia de pesquisa utilizada para desenvolver a proposta do sistema de apoio para gestão de competências baseada em agentes.

1.1 APRESENTAÇÃO

A disciplina de Engenharia de Produção surgiu no Brasil no ano de 1957 na Escola Politécnica da Universidade de São Paulo (FAÉ et Al, 2004, p. 01), dedicando-se nesta época à formação de profissionais exclusivamente para postos de trabalho operacionais no setor produtivo.

Atualmente é considerada área multidisciplinar, com particularidades que lhe permite atuar na fronteira entre as Engenharias, Administração e Tecnologia (AMARAL et. Al, 2004, p. 02). Segundo a Associação Brasileira de Engenharia de Produção, a Engenharia de Produção é formada por profissionais de diversos segmentos como a Gerência da Produção, Qualidade, Pesquisa Operacional, Gestão Ambiental e Sistemas de Informação, dentre várias outras. A Figura 1 demonstra o conhecimento (Kn) disperso entre as diversas áreas.

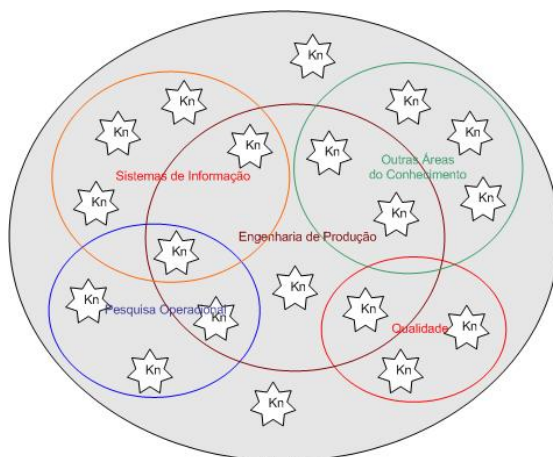


Figura 1 – Composição da Engenharia de Produção

Devido a esta diversidade funcional, mostra-se relevante a utilização de uma ferramenta que auxilie na identificação de conhecimentos dos egressos. Estas informações

podem auxiliar na verificação dos trabalhos desenvolvidos durante o curso de pós-graduação e, se na seqüência, estes convergem com os interesses da Engenharia de Produção, conforme demonstra a Figura 2.

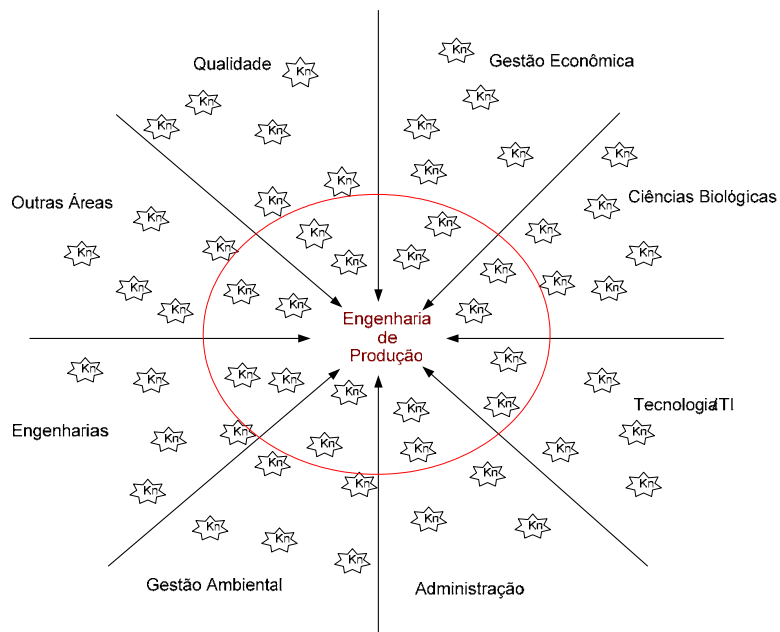


Figura 2 – Convergência das diversas áreas para a Engenharia de Produção

O contexto da problemática discutida anteriormente sobre as fronteiras da Engenharia de Produção e suas respectivas áreas do conhecimento, permite perceber a seguinte questão de pesquisa:

“É viável a construção de uma ferramenta informatizada com tecnologia inovadora, para que as Instituições de Ensino de Pós-Graduação em Engenharia de Produção buscassem indícios de competências nos currículos de seus egressos, com o intuito de auxiliar o gestor na manutenção de uma grade curricular atualizada com as necessidades do mercado?”

1.2 JUSTIFICATIVA E MOTIVAÇÃO

Atualmente, existem vários sistemas informatizados disponíveis na rede Internet que permitem a gerência de informações profissionais. Pode-se citar portais como o Catho (<http://www.catho.com.br>) e o Centro de Solidariedade ao Trabalhador (<http://www.cst.org.br>). Porém, estas ferramentas são destinadas ao mercado corporativo,

ou seja, as informações curriculares são fornecidas com o intuito de pleitear um posto de trabalho.

Para a comunidade científica e acadêmica foi criado o Sistema Lattes, que é composto pelo Portal da Plataforma Lattes, pelo sistema de cadastro de currículos *off-line* conhecido como Currículo Lattes e pelo sistema Lattes Extrator dentre outros. O sistema Lattes Extrator é o módulo que permite às instituições de ensino licenciadas, coletar informações curriculares dos seus pesquisadores, em lotes, diretamente do banco de dados de Currículos Lattes do CNPq. Todavia, para o uso deste módulo do sistema, existe custo.

O sistema de cadastro *off-line* de Currículos Lattes foi projetado a partir da versão 1.4 com o intuito de permitir intercâmbio de dados entre os sistemas de gestão das I&E&P e o sistema de Currículos Lattes. Esta troca é realizada através de um padrão aberto de estruturação de dados chamado XML (*Extended Markup Language*) conforme consta em (PORTAL LATTES. Histórico do Currículo Lattes, 2007).

Devido ao fato de que o Currículo Lattes é uma ferramenta para coleta de dados curriculares comum no meio científico, o estudo de viabilidade de seu uso para a coleta de informações para posterior mapeamento em sistema informatizado de apoio onde estas informações curriculares podem ser analisadas e quantificadas.

O contexto deste trabalho de pesquisa propõe o desenvolvimento de ferramentas informatizadas, baseadas em Sistemas Agentes, que permitam mensurar os indicadores que correspondam às necessidades apresentadas, descritas na forma dos objetivos deste trabalho científico.

A criação de sistemas informatizados que possibilitem a análise das competências dos egressos, devido a estes já constarem no banco de dados do Sistema de Currículos Lattes, pode auxiliar o gestor no processo de estruturação do curso nas diretrizes necessárias à Engenharia de Produção, bem como potencializar a manutenção da grade curricular através da observação das necessidades do mercado.

1.3 OBJETIVOS

O objetivo da pesquisa é o desenvolvimento e ensaio de um sistema protótipo que permita auxiliar na busca de informações curriculares para o posterior mapeamento de competências, em egressos de programas de Pós-Graduação em Engenharia de Produção (PPGEP) utilizando-se como instrumento de aquisição de informações o sistema Currículo Lattes.

O desenvolvimento desta proposta desmembra-se nos seguintes objetivos específicos:

- avaliar os requisitos necessários e as implicações na organização de um sistema para apoiar as atividades de gestão competências;
- analisar a consistência da linguagem C# frente das principais linguagens de programação utilizadas atualmente no mercado, para verificar a melhor opção a ser utilizada no desenvolvimento da ferramenta;
- avaliar o potencial e viabilidade da plataforma JADE para desenvolvimento de sistemas agentes em um ambiente de programação .Net C#;
- planejar e desenvolver um sistema protótipo para testar a viabilidade da utilização do JADE em ambiente .Net para o desenvolvimento de sistemas de gestão de competências.
- ensaiar o sistema desenvolvido utilizando amostras de currículos, com o objetivo de testar suas potencialidades e limitações.

1.4 METODOLOGIA DE PESQUISA

Nesta seção, é apresentada a metodologia utilizada na construção deste trabalho.

Buscando explorar de uma melhor forma o tema “Competências”, este trabalho se inicia pela pesquisa bibliográfica de seus conceitos, apresentando as bases históricas e a evolução do termo.

Com o intuito de se verificar estruturas metodológicas que permitam auxiliar na busca por indicadores de competências, são apresentadas na seqüência algumas metodologias que se propõe a auxiliar o gestor no processo de análise de dados onde se buscam competências.

Como um dos objetivos deste trabalho é o desenvolvimento de uma ferramenta informatizada, é efetuada a pesquisa das principais linguagens de programação e mecanismos de armazenamento de dados. Esta pesquisa busca verificar quais são as tecnologias mais relevantes a serem utilizadas no desenvolvimento da aplicação.

Para verificar a possibilidade de se utilizar uma solução já existente como ferramenta de coleta de dados curriculares para posterior tratamento e busca de indícios de competências, foram explorados alguns dos sistemas informatizados encontrados no

mercado e na academia, que se propõe a efetuar o armazenamento de informações curriculares, bem como a gestão de competências.

Como o foco deste trabalho é o desenvolvimento de uma ferramenta baseada em sistemas agentes, são apresentados os conceitos de agentes de software e o seu emprego no desenvolvimento de sistemas informatizados. É abordado o conjunto de bibliotecas para sistemas agentes JADE, sendo que esta plataforma será utilizada posteriormente na construção do protótipo apresentado por este trabalho.

Com base na análise das teorias e tecnologias apresentadas e discutidas no referencial teórico, é apresentada uma proposta de sistema agentes para busca de indícios de competências em egressos dos Programas de Pós-Graduação em Engenharia de Produção, através do uso da tecnologia agentes utilizando o ambiente JADE.

O sistema proposto foi desenvolvido e ensaiado com os dados obtidos junto a egressos do Programa de Pós-Graduação em Engenharia de Produção (PPGEP) da Universidade Tecnológica Federal do Paraná (UTFPR).

1.5 ESTRUTURA DO TRABALHO

Este trabalho de pesquisa está dividido em cinco capítulos. O primeiro capítulo proporciona uma visão geral deste trabalho de pesquisa, apresentando o contexto deste trabalho e sua justificativa, os objetivos, a metodologia de pesquisa e a estrutura do trabalho que é descrita neste item.

No segundo capítulo, encontra-se a revisão bibliográfica onde são fundamentados os conceitos estudados no decorrer deste trabalho. São tratadas as bases históricas e evolução do termo competências, a teoria da complexidade e sua visão na busca por indicadores de competências e algumas das metodologias que se propõe a auxiliar no processo de gestão de competências. Em seguida, são apresentadas as linguagens de programação e bases de dados utilizadas na construção de sistemas informatizados, abordando os conceitos de linguagens de programação e bases de dados. Após, são analisados os sistemas informatizados encontrados no mercado que auxiliam em processos de gestão de competências. Esta seção finaliza com a explanação da tecnologia agentes apoiada pelo uso do *framework* JADE, para a construção de sistemas informatizados.

O terceiro capítulo apresenta a proposta para a construção de um sistema para busca de indícios de competências de egressos dos programas de Pós Graduação em

Engenharia de Produção, codificado em linguagem C# .NET juntamente com o *framework* agentes JADE.

No quarto capítulo, é efetuado o ensaio do sistema construído, onde é verificada a viabilidade técnica da ferramenta e são tecidas considerações quanto à sua usabilidade, bem como o processo de construção.

O quinto capítulo apresenta as conclusões deste trabalho de pesquisa, onde foram empreendidas discussões que reafirmaram o contexto deste trabalho quanto aos objetivos propostos, limitações da ferramenta e sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo discorrerá sobre a pesquisa bibliográfica dos assuntos pertinentes à construção da ferramenta informatizada, que buscará apontar indicadores de competências nos currículos de egressos dos cursos de Pós-Graduação em Engenharia de Produção.

Neste contexto, serão empreendidas discussões quanto à origem e evolução do termo competência, sobre os sistemas informatizados existentes que prometem gerir competências e, também serão tratadas, as principais tecnologias utilizadas na construção de ferramentas informatizadas, buscando sedimentar os conceitos que serão mais tarde aplicados durante a construção do protótipo e no ensaio.

Como o escopo deste trabalho visa a elaboração de um sistema informatizado que permita a organização de indicadores de competências, este capítulo têm por objetivo em primeiro momento, efetuar um estudo teórico sobre o tema competências.

Em um segundo momento, serão apresentados alguns sistemas informatizados comerciais e discussões acadêmicas, que demonstram diferentes formas de efetuar a busca e organização de competências individuais.

Ao final deste referencial teórico, será efetuado o estudo das tecnologias que podem ser utilizadas na construção de sistemas informatizados, buscando verificar qual é a melhor técnica a ser utilizada na construção do sistema, que deverá efetuar simulações durante o ensaio da ferramenta.

2.1 CONCEITO DE COMPETÊNCIA

Devido ao advento da globalização, o fator de diferenciação buscado pelas organizações baseia-se cada vez mais em elementos tácitos Dergint (2001, p.02). Neste contexto, devido a vários fatores tais como a ubiquidade da informação e do conhecimento explícito disponível, e a tendência de tratá-los como *commodities* Maskell (1997, p.02) Dergint (1999, p.37) Lundvall (2000, p. 06-07), busca-se a geração de diferencial mercadológico pela capacidade de articulação de fatores que, são pouco móveis como o conhecimento tácito, as competências e os valores sociais.

Frente a este cenário, o estudo da capacidade de construção organizacional de competências permite que a empresa obtenha, em termos organizacionais de identificação de oportunidades, um diferencial competitivo proporcionando a busca de uma melhor forma

de estimular, incentivar e preservar estas propriedades na empresa. Rocha Neto (2004, p.78) afirma que conhecimentos e competências são os ativos mais importantes das empresas.

Indivíduos e organizações que solucionam problemas em conjunto têm a possibilidade de partilhar o conhecimento original do parceiro, do mesmo modo como terão partilhado o novo conhecimento tácito gerado em conjunto. O aprendizado partilhado é a chave para o conhecimento tácito gerado em conjunto (LUNDVALL, 2001, p.202).

Desta forma, o estudo das competências se tornou à luz dos estudos de inovação, um caminho que vem “encarecendo” em importância de tal forma que, as empresas necessitam criar mecanismos para estimular processos competitivos em suas estruturas. Neste sentido, o conceito de competência como vantagem competitiva tem levado as empresas a identificarem novas oportunidades de negócios e a utilizar estrategicamente os seus ativos intelectuais (GIACOBBO, 2004, p.01).

Os ativos intangíveis são difíceis de serem copiados/imitados e se revelam de apropriação improvável por parte de terceiros, pois dependem de culturas organizacionais, trajetórias históricas e aprendizagens, constituindo identidades distintas e ativos que conferem vantagem competitiva. Rocha Neto (2004, p. 78)

Conforme fundamentado pelos autores, observa-se que as competências são elementos essenciais para todas as atividades humanas, tanto no âmbito individual quanto no coletivo, sendo muito associadas ao processo de inovação no contexto empresarial.

O fato de o conhecimento que é crucialmente importante para o êxito na economia ser o tácito, tem importantes implicações para os princípios eleitos para a política de inovação. (LUNDVALL, 2001, p. 203)

No auxílio à manutenção da competitividade pelas empresas, as competências são diferenciais que podem ser observadas no contexto da criação de novos produtos e serviços, buscando atender a um público cada vez mais exigente e seletivo de forma diferenciada.

A globalização dos mercados, as inovações tecnológicas, e o aumento da competitividade trazem, como consequência, a implantação nas empresas de programas para a maximização de seus resultados. Diante desse contexto, as empresas estão aprendendo que a única maneira de construir a verdadeira vantagem

competitiva é por intermédio do capital humano. (BARRET, 2000) apud Oliveira (2004, p. 01)

Na busca por novas formas de conquistar espaço neste novo modelo de mercado, o colaborador possui papel muito importante, avaliado pela possibilidade que este tem de construir a *expertise* necessária para auxiliar a organização no cumprimento de suas metas. Rocha Neto (2004, p.78) relembra que para algumas organizações, o capital intelectual tornou-se a vantagem competitiva que faz realmente a diferença, dados que as demais condições podem ser facilmente obtidas no mercado. Estas “habilidades” emergentes podem indicar a existência de determinadas competências.

No escopo deste trabalho, pretende-se verificar a evolução do termo competência e suas características sob a óptica de diferentes autores, buscando compreender de uma melhor forma os aspectos apresentados por cada vertente. Isto é importante para fundamentar a construção de um quadro que dê suporte ao tratamento e avaliação de conhecimentos que indiquem competências potenciais¹.

2.1.1 Bases históricas e evolução do termo competência

A origem da palavra competência segundo Rabechini Júnior (2001, p.03) vem do latim, *competere*. Seu significado pode ser observado, inicialmente com a decomposição da palavra em latim *com*, cujo significado é conjunto e *petere*, que significa esforço.

Segundo Cury (2004, p. 01), historicamente, o termo competência começou a ser utilizado por juristas no final da Idade Média, para determinar se um tribunal seria capaz para executar determinado tipo de julgamento. Através dele, era atribuído à um indivíduo ou instituição a “competência” para por em prática a execução de determinado procedimento.

Nos estudos acadêmicos pertinentes ao tema competência, verifica-se que os autores fundamentam o conceito sob duas perspectivas principais, a americana e a francesa. A americana possui uma visão mais comportamentalista, focada nas capacidades dos indivíduos. A francesa observa o indivíduo sob uma óptica construtivista, vinculando trabalho e educação (PEREIRA, 2005, p. 02).

¹ Ver item 2.1

Concepções comportamentalista e construtivista

Na vertente americana, o pesquisador David Clarence McClelland é um dos primeiros a tratar o tema competência na forma diretamente ligada ao comportamento do indivíduo. Seu trabalho intitulado “*Testing for Competence Rather Than for ‘Intelligence’*” McClelland, apresentou um estudo comparativo entre profissionais considerados “mais qualificados” e “menos qualificados” para o mercado de trabalho, verificando quais conseguiram bons empregos e outros que não conseguiram tal projeção profissional (McCLELLAND, 1973, p.01). Neste trabalho, a coleta dos dados foi estruturada através de testes aptidão (McCLELLAND, 1973, p.05), mostrando que todos os selecionados responderam os testes, porém no momento de colocar os conhecimentos em prática, uns conseguem e outros não. Este trabalho conclui que, deveria ser estruturada uma avaliação de competências como alternativa aos testes de inteligência do tipo acadêmico, os quais falham em mostrar o sucesso de desempenho do indivíduo (McCLELLAND, 1973, p.11-13) (Flink, 2005, p. 01).

NONAKA (1997, p. 28-29) também observa a competência sob a óptica comportamentalista afirmando que seu desenvolvimento depende da capacidade de compreensão e da intuição do indivíduo. De Re (2002) *apud* Dantas *et al* (2005, p. 02) afirma que é muito importante a observação de algumas características comportamentais, pois elas podem comprometer o sucesso do indivíduo, influenciando nas aspirações e caracterização da sua visão de mundo.

A visão de Chiavenato (2004, p. 53-54) sustenta que a competência é um sinônimo de talento humano, resultante do acúmulo dos conhecimentos e habilidades. Este talento é composto por conhecimentos, *expertise*, poder de inovação e habilidades dos colaboradores aliados aos valores, cultura e filosofia da empresa Colauto (2004, p. 05), Edvinsson *et al* (1998) *apud* Gracioli (2005, p. 03), Lima (2003, p.07).

Para Prahalad (1999) *apud* Venturini (2005, p. 02), a busca por novas competências nas organizações precisam estar voltadas não só para o aprendizado individual, o que inclui o desenvolvimento de capacidade analítica. Faz-se necessário observar o comportamento e as habilidades especiais dos indivíduos.

Na vertente francesa destaca-se Thomas Durand com o artigo “*Forms of Incompetence*” Durand (1998) onde é proposto um modelo construtivista bastante conhecido de competência fundamentado em conhecimentos que são as informações já assimiladas pelo indivíduo, habilidades como capacidade de aplicar o conhecimento adquirido e atitudes para converter conhecimentos e habilidades em ações agregando valor social à pessoa,

sendo que esta competência não se limita a um conhecimento específico ou estado (SILVA, 2003, p.02) (RABAGLIO *apud* CARLETTO *et al* 2005, p. 03) (FLEURY, 2000, p.53).

A competência é um cluster de conhecimentos, habilidades e atitudes relacionados que afetam a maior parte de um papel profissional ou responsabilidade, que se correlaciona com a performance desse papel ou responsabilidade, que possa ser medido contra parâmetros “bem aceitos” e que pode ser melhorado através de treinamento (PARRY 1996 , p.50 apud DUTRA 2002, p.21).

Para Le Boterf (1995), a competência é construída através da formação educacional e experiência profissional do indivíduo, conduzindo-o a atuar de forma responsável. Este autor também observa o estabelecimento da competência na forma construtivista, onde ela emerge como resultado do produto de aprendizagens, formação e sistema de avaliações proporcionando ao indivíduo em um determinado contexto, executar determinada tarefa demonstrando sua capacidade.

Refletindo sob a mesma óptica, Sveiby (1998, p.42) complementa que a competência é o resultado do acúmulo de experiências que um indivíduo adquire durante sua existência. Estes conhecimentos reunidos estariam em processo de construção constante, o que lhe permite executar ações, tomar decisões, ter determinado comportamento perante uma situação e gerar resultados.

Habilidades e conhecimentos são transformados em competências pelos verbos, mobilizar, assimilar, participar, comprometer Zarifian (2001) apud GIACOBBO (2001, p. 02).

Para Lévy (1996, p. 60), Resende (2000, p. 32), Zarifian *apud* Fleury (1999, p. 190) e Harzalah *et al* (2006, p. 1-3), a competência é considerada *expertise* de situações que se apóiam sobre a transformação dos conhecimentos adquiridos. É referenciada como capacidade contínua de aprender e inovar, podendo ser atualizada de maneira imprevisível em contextos variáveis, transformando conhecimentos, aptidões, habilidades, interesses, vontade, etc. em resultados práticos.

Observa-se no modelo americano que a competência é formada em função do comportamento que o indivíduo assume diante de determinada situação, o que lhe permite através de seu conjunto de qualificações, executar um trabalho com um nível superior de desempenho. Neste sentido, este modelo conduz a observar a formação da competência como resultado da intersecção entre os conhecimentos, habilidades e atitudes conduzidos à formação pela influência do comportamento humano conforme demonstra a Figura 3.

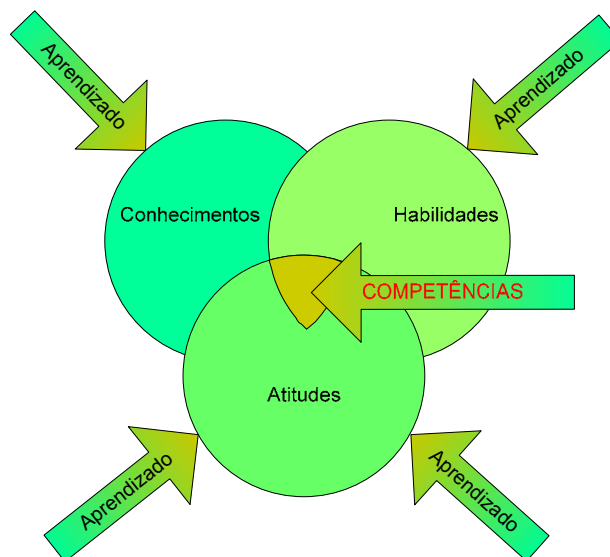


Figura 3 – Formação da competência sob a ótica americana. Fonte: Durand (1998)

Já na concepção francesa a competência é construída através de aquisição de conhecimentos (processo de aprendizagem). Neste modelo a habilidade apresenta-se como experiência caracterizando-se pela capacidade que o indivíduo possui de aplicar o conhecimento adquirido. A atitude é expressa como sendo a “vontade de fazer”.

Analisado sob este aspecto, a competência é sustentada por processos de conhecimento que envolvem aprendizado e, mapeada sobre as atitudes, habilidades e conhecimentos apresenta-se como uma propriedade emergente, conforme demonstra a Figura 4.

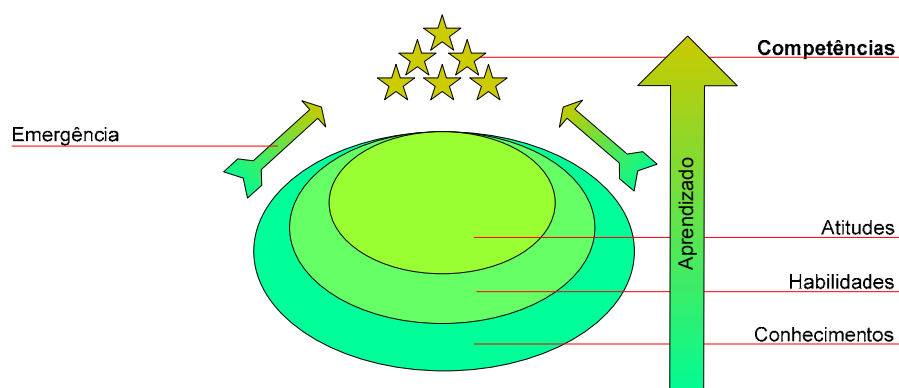


Figura 4 – Formação da competência sob a ótica francesa. Fonte: Durand (1998)

Embora as figuras apresentadas anteriormente demonstrarem uma representação gráfica do processo de desenvolvimento de competências, dá-se a sensação que as

competências são um subconjunto menor. Todavia, analisam-se as competências não como em menor proporção frente a outras características do indivíduo, mas como sendo uma propriedade emergente das estruturas anteriores, onde elas se manifestam.

Normalmente em um contexto sócio-econômico a manifestação da competência individual passa pela manifestação de competências de outras pessoas². Fleury (1999, p. 192-193) afirma que estas competências externas podem apresentar-se nas seguintes formas:

Competências Técnicas ou Profissionais: são específicas de um cargo ou tarefa e segundo Freire (2001, p.08) e Carletto (2005, p. 03) são fundamentadas em conhecimentos sistematizados e tangíveis, dando a possibilidade de que o seu processo de aprendizado ocorra de forma mais fácil;

Competências Sociais: Rocha Neto (2004, p. 79) afirma que as competências sociais referem-se à capacidade de escutar ativo – indagar para entender as inquietações dos interlocutores, inclusive para desaprender, prometer/cumprir com responsabilidade, emitir juízos com fundamentos, ensejar enfoques múltiplos, além da apreciação de questões relevantes, a partir de diferentes pontos de vista, e coordenar ações, de forma sistemática, comprometida e responsável envolvendo conseqüentemente a capacidade de dizer sim e não.

Competências de Negócio: são relacionadas à compreensão do negócio bem como seus objetivos nos contextos mercadológicos como o ambiente político ao qual a organização ou sociedade encontra-se imersa. Freire (2001, p.08) complementa que este tipo de competência permite identificar oportunidades, ameaças e pontos fortes e fracos.

É importante ressaltar que no contexto deste trabalho, não foi esgotada a tipologia de competências externas encontrada hoje na literatura, sendo que existem autores que as observam sob outras óticas. Estas competências externas adicionais não foram citadas neste trabalho devido ao fato de não fazerem parte do escopo a ser tratado na análise dos dados pela ferramenta informatizada, onde serão buscados indicadores de competências técnicas ou profissionais.

² Competência é uma característica que se manifesta no indivíduo e é intrínseca dele. No contexto deste trabalho entende-se competências como coletiva ou organizacional como a manifestação de diversas competências individuais.

Desta forma, os autores demonstram que as competências não ficam limitadas a um conjunto de conhecimentos detidos pelo indivíduo e nem intrínsecos a um procedimento ou tarefa.

Observa-se que a manifestação de competências como propriedades emergentes de vários níveis como o profissional, o social e do negócio, são referenciados por Dubar (1998, p. 04) como “relações sociais complexas”. Esta propriedade emerge quando as competências encaixam-se em determinado contexto, sozinha ela não possui dimensão suficiente para articular o conhecimento em uma função.

Qualificação e competência são, portanto, noções inteiramente relativas aos modos de articulação destas relações sociais que definem a profissionalidade num contexto dito “social” DUBAR (1998, p.04).

Neste cenário, a teoria da complexidade têm possibilidade de auxiliar na busca de indicadores de competências pelo fato de que ela auxilia na compreensão do funcionamento de sistemas dinâmicos e não lineares, onde a visão cartesiana mostra-se insuficiente no entendimento das relações existentes entre fatores tácitos como conhecimentos e competências.

Não há como não “enxergar” que os conceitos de complexidade, organização, autopoiese, pluralidade, informação, sinergia, entre outros, não estejam também pertinentes às relações sociais (ROCHA NETO 2003, p. 89).

2.1.2 Competências e a Teoria da Complexidade

A visão holística, que o dicionário Aurélio (1999) define como a “Teoria segundo a qual o homem é um todo indivisível, e que não pode ser explicado pelos seus distintos componentes (físico, psicológico ou psíquico), considerados separadamente” é um padrão de pensamento antagônico à visão sistêmica (ROCHA NETO, 2004, p. 24) (MORIN, 2005, p.17 p.20), sendo que ambos podem ser utilizados para investigar objetos e suas propriedades. Enquanto a visão holística observa o todo, a abordagem sistêmica preocupa-se em observar as partes e suas relações existentes com outras dimensões, que podem extrapolar o objeto em estudo (ROCHA NETO, 2004, p.13). Por exemplo, a visão holística de uma obra de arte percebe-a por suas propriedades quanto à estética, forma, cor, valor, etc. enquanto na visão sistêmica pode-se explorar sua importância histórica, seu elo com o

artista e com suas outras obras e fases, buscando distinguir sinergias globais ou emergentes, que podem não estar presentes em nenhuma de suas partes.

Rocha Neto (2003, p.71-72) demonstra que historicamente, a construção do paradigma sistêmico ocorreu em três atos, iniciando ao final de 1940, com a contribuição das ciências biológicas no entendimento dos sistemas de controles das máquinas e o seu auxílio recíproco à compreensão dos seres vivos. Em seu segundo momento, houve a tentativa frustrada de unificar as disciplinas científicas, contudo foi desenvolvida uma poderosa ferramenta de observação, a Teoria Geral dos Sistemas, que segundo Morin (2005, p. 19) foi iniciada por von Bertalanffy em uma reflexão sobre a biologia a partir dos anos 50. O terceiro momento culminou na fertilização cruzada de conceitos típicos da abordagem sistêmica, resultando na interação das disciplinas com a introdução da inteligência artificial, das técnicas de reconhecimento de padrões, da biônica e da robótica, que daria origem em um quarto momento ao nascimento da Cibernética, atribuída ao cientista americano Norbert Wiener em 1948.

A possibilidade de que máquinas pudessem imitar procedimentos humanos e funções próprias de seres vivos contribuiu para elucidar alguns processos cerebrais. Bem como o desenvolvimento da inteligência artificial, da biônica e da robótica (ROCHA NETO 2003, p.74).

Rocha Neto (2003, p.69-70) e Morin (2005, 19-20) definem a idéia de sistema como um conjunto de elementos que formam um todo indivisível, interagindo dinamicamente e organizando-se com um propósito. Pode-se citar como exemplo uma pessoa, que é um sistema formado por células – sistemas de si mesmos, que se conectam para formar órgãos, que realizam processos específicos para o funcionamento do corpo, unidade vivencial de cada pessoa, que se conecta no ambiente de uma família, que vive em uma cidade, em um país e assim por diante.

Estes sistemas nas palavras de Rocha Neto (2004, p. 14; p. 70), podem ser simples ou complexos, de acordo com os graus de ordem e de complexidade que apresentam. Quando os sistemas são complexos, pode existir sinergia entre as partes, resultando em respostas finais maiores que a soma dos resultados das partes do sistema. Caso as relações mútuas todo/partes e partes/todo não se comprovem, não existe relação de pertinência entre as partes configurando uma “unidade sistêmica”.

O termo complexidade é de difícil conceituação, existindo muitas definições, sendo que algumas enfatizam a complexidade do

comportamento do sistema, outras enfatizam a estrutura interna do sistema, seu funcionamento. Por outro lado, o conceito complexidade pode ser encontrado em vários campos, desde os sistemas naturais, representados pelos sistemas biológicos, físicos e químicos aos sistemas artificiais, tais como sistemas computacionais e estruturas organizacionais (LEITE, 2004, p.02).

Segundo Morin (2005, p. 13), a complexidade pode ser observada como um tecido constituído por matérias heterogêneas inseparavelmente associadas onde são observados acontecimentos, ações, interações, retroações, determinações e acasos resultantes de cadeias de fenômenos.

A complexidade surge, é verdade, lá onde o pensamento simplificador falha, mas ela integra em si tudo o que põe ordem, clareza, distinção e precisão no conhecimento MORIN (2005, p. 6).

Rocha Neto (2003, p. 69; p. 90), Lynaldo (2006, p.61) Morin (2005, p. 33-37) fundamentam que as principais características que compõe a complexidade são:

- Incerteza ou indeterminação: idéia de que não se pode controlar os acontecimentos futuros.
- Autopoiese e auto-organização: capacidade dos seres vivos ou sociais de gerar e manter suas próprias organizações.
- Complexidade: os sistemas compreendem relações de interdependência recíprocas e indeterminadas.
- Sinergia: as respostas finais apresentam-se maiores que a soma das partes, resultante de relações não lineares entre as partes componentes de um sistema. Nos sistemas lineares as respostas globais são únicas e iguais às somas das contribuições de todas as partes.

A complexidade de um sistema, segundo Rocha Neto (2004, p. 14-21) pode ser medida pela diversidade e características de seus elementos tais como grupabilidade, separabilidade, explicabilidade, partes e conexões; também pela sensibilidade apresentada em relação às condições iniciais e pelas diferentes interações entre as partes.

Segundo SATO (2004, p. 98) e DUBAR (1998, p.04), pode-se construir uma conexão intuitiva entre os conceitos de complexidade e de compreensão, pois a compreensão possibilita o indivíduo a utilizar, modificar, controlar ou observar algo novo

independente do contexto ao qual está inserido. Esta compreensão pode ser alcançada de várias formas como a classificação, descrição, dentre outras.

Neste contexto, observa-se que as competências dos indivíduos podem apresentar-se como propriedades emergentes imersas em um sistema complexo, possuindo características que têm a possibilidade de interagir de forma não-linear. El-Hani (2000, p. 203) define propriedade emergente como característica que deve potencialmente estar presente apenas em objetos complexos, não estando presente em qualquer uma das partes do objeto e sendo distinta de qualquer propriedade estrutural do objeto em estudo.

A não-linearidade de um sistema implica em que a resposta não é proporcional ao estímulo aplicado, por isso não se pode prever seu comportamento futuro com exatidão Bauer (1999) apud (SATO 2004, p. 100).

Neste sentido, podem existir determinadas competências que só podem ser articuladas quando existirem relações sociais, manifestando-se através de confiança ou trajetória histórica entre os indivíduos, outras, podem ser estimuladas pelo ambiente. Estas propriedades estabelecem relações de forma interdependente, significando que as propriedades do todo podem não estar presentes em cada uma de suas partes e que a ausência de uma das partes tem grande impacto no todo.

Qualquer conhecimento opera por seleção de dados significativos e rejeição de dados não significativos: separa, (distingue ou disjunta) e une (associa e identifica); hierarquiza (o principal, o secundário) e centraliza (em função de um núcleo de noções-chaves) (MORIN, 2005, p. 10).

2.1.3 Visão da Teoria da Complexidade na busca por indicadores de competências

Analisando sob a ótica da Teoria da Complexidade, que possibilita auxiliar de uma melhor forma no entendimento da subjetividade existente nos processos sociais, verifica-se que os conhecimentos explícitos podem ser apresentados como propriedades do indivíduo, e suas relações com outras propriedades e processos podem evidenciar indicadores que reflitam sua competência.

“[...] por isso o conhecimento necessita ordenar os fenômenos rechaçando a desordem, afastar o incerto, isto é, selecionar os elementos da ordem e da certeza, precisar, clarificar, distinguir...” (MORIN, 2005, p. 13-14)

Como o escopo deste trabalho tem por objetivo buscar indícios de competências em egressos de cursos de pós-graduação em Engenharia de Produção, evidencia-se a tentativa de buscar homologias entre os sistemas biológicos e as características humanas, através da investigação e entendimento dos processos sociais existentes no ambiente onde os egressos encontram-se imersos.

Nesta busca por indicadores de competências, podem emergir cenários que sugiram a existência de interação entre as medidas sistêmicas, gerando sinergia e demonstrando as propriedades que não estão contidas nas partes. Isso pode acontecer devido a relações no ambiente onde o egresso esteve inserido, como por exemplo, treinamentos e cursos que podem resultar em aprovações de artigos em congressos, bolsas de estudo ou prêmios de pesquisa concedidos por órgãos de fomento.

É importante entender a competência e a sua dimensão complexa, para que se possa verificar que os sistemas informatizados não tem a capacidade de efetuar estas correlações. Neste contexto, procurar-se-á efetuar um levantamento de indicadores que venham a auxiliar o gestor em sua decisão.

Neste processo de busca de elementos tangíveis existem simplificações, sendo que muitas delas podem ser questionadas em função do quadro teórico tratado até o presente momento. Autores como Chiavenato (2004) procuram quantificar e qualificar as competências sob o aspecto dos elementos tangíveis (explícitos). Seguindo outra linha Cooper (1997) dentre outros, buscou observar e quantificar os aspectos intangíveis (tácitos) através de modelos como o EQMAP³, sugerido pelo próprio autor em questão na busca de contabilizar a inteligência emocional do indivíduo.

Porém, como os elementos que deverão ser tratados neste trabalho são os exclusivamente tangíveis de avaliação, buscou-se gerar mecanismos informatizados modulares que trabalhem de maneira autônoma, com base nos estudos dos autores supracitados. Isto possibilita o tratamento de uma melhor forma de grandes volumes de informação tendo a possibilidade de até, em versões mais sofisticadas, efetuar correlações maiores, que os sistemas convencionais não informatizados são incapazes de apresentar.

³ Questionário que procura extrair através de perguntas objetivas, um valor que demonstre a “inteligência emocional” do entrevistado. O referido autor sugere o uso deste indicador para demonstrar conhecimentos tácitos da pessoa.

2.1.4 Metodologias para busca de indicadores de competências

No escopo deste capítulo que é o de discutir os conceitos e evolução do termo competência, este item procurará tratar algumas das metodologias existentes no mercado que se propõe efetuar a busca por indicadores de competências. Neste contexto, na busca por indicadores de competências aqui discutida, são explicitadas metodologias que apresentam formas de tratar as competências técnicas ou profissionais dos colaboradores. As metodologias aqui propostas podem ser utilizadas em processos de entrevistas, avaliações por questionários, análises curriculares e outros.

Estas metodologias descritas na seqüência, elaboradas por Chiavenato (2004, p.130-140) e aplicados de forma direta e indireta pelas empresas Siemens e Efix têm permitido a avaliação do conhecimento disponível, através da busca por elementos tangíveis de avaliação na forma de competências técnicas ou profissionais, discutidos anteriormente. Elementos estes que mais tarde serão demonstrados neste trabalho na forma de currículos que são elementos tangíveis de avaliação, onde serão buscadas correlações com elementos não explícitos normalmente tácitos como conhecimentos e competências.

2.1.4.1 Metodologia proposta por Chiavenato (2004)

Na busca por indicadores que tenham a possibilidade de representar as competências de um indivíduo, Chiavenato (2004, p.131-132) apresenta dentre outros processos distintos utilizados para a análise de características individuais, um processo chamado de "Processo de Comparação". Na tentativa de melhor explicar o referido processo, o autor faz uma analogia a comparação de duas variáveis, onde de um lado representam-se os requisitos buscados e de outro as características dos indivíduos, conforme demonstrado na Figura 5. A primeira variável (x) é fornecida através da descrição de características inerentes ao cargo ou tarefa a serem desempenhadas ou competências necessárias, e a segunda (y) é obtida através da aplicação de técnicas de seleção.

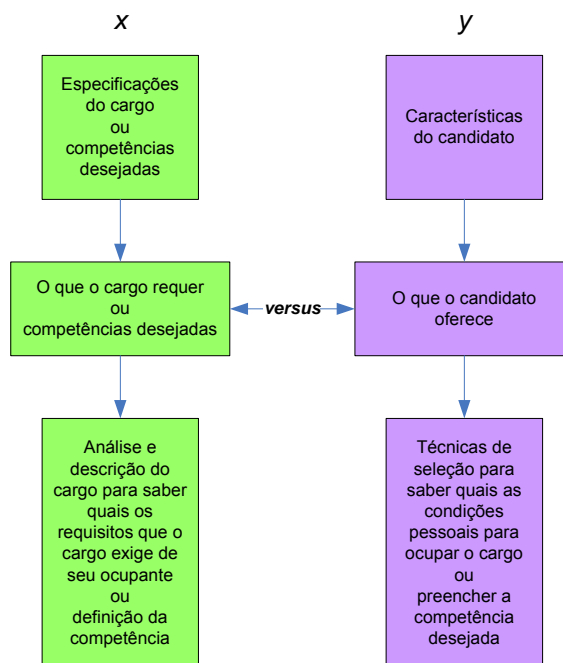


Figura 5 – Seleção de Pessoal. Fonte: Chiavenato (2004, p. 132)

Se de um lado temos o cargo a ser preenchido ou as competências desejadas, temos de outro, candidatos profundamente diferentes entre si, disputando a mesma posição. Nestes termos, a seleção passa a ser configurada basicamente como um processo de comparação e decisão. CHIAVENATO (2004, p.132)

Desta forma, Chiavenato (2004, p. 132) demonstra que, quando (x) é maior que (y) , o indivíduo não possui condições de ocupar o cargo ou executar a tarefa, enquanto se a situação for inversa, ele possui mais do que as condições necessárias ao cargo, sendo considerado *superqualificado* para a função. Este cenário demonstra que a situação ideal é quando a variável (x) iguala-se⁴ à (y) .

Conforme demonstrado no cenário acima, o autor fundamenta que, para possibilitar esta comparação, é necessário descrever qual a competência buscada,

⁴ Como são analisadas diversas características e possibilidades, o processo de comparação não se concentra em um único ponto de igualdade, mas em uma faixa de aceitação imposta através da definição de limites de tolerância, o que permite uma certa flexibilidade ao entorno do ponto ideal buscado (CHIAVENATO 2004, p.132).

explicitando-a. No caso, em (x) deverão ser descritas estas competências buscadas e em (y), as competências oferecidas que podem estar expressas na forma de um currículo, que caracterizam um indivíduo. Neste contexto, pretende-se buscar em currículos, indicadores que possibilitem ao gestor, avaliar competências técnicas atuais e também, indicar através de uma curva de tendência um prognóstico futuro que possa demonstrar como estas variáveis irão comportar-se em longo prazo.

Observando o processo de busca por indicadores de competências seguindo o modelo proposto por Chiavenato (2004, p.135), têm-se o processo de seleção como um processo de comparação e escolha, sendo que o mesmo precisa apoiar-se em padrões e critérios para que os resultados sejam válidos.

Neste sentido, Chiavenato (2004, p. 137-138) afirma que, quando se têm um processo seletivo fundamentado em competências, a coleta de dados se limita à definição destas competências. Quanto melhor for definida a competência, mais ela se torna um instrumento viável de medida para comparar indivíduos.

2.1.4.2 Metodologia de avaliação de competências SCoM/Siemens

A metodologia SCoM, *Score Competence Management* nasceu em 1999 na Alemanha na academia Werner Von Siemens, da multinacional Siemens. Sua implantação teve início em 2001 através de uma ferramenta informatizada interna da empresa que era conhecida por *Hestia*. Esta ferramenta tinha por objetivo possibilitar a retenção do conhecimento interno da Siemens, através do armazenamento do conhecimento tácito dos funcionários, que uma vez convertido em explícito poderia ser armazenado em bases digitais, possibilitando a geração de bases de conhecimento que pudessem ser consultadas pelos colaboradores internos da empresa.

Mais tarde, houve um esforço da empresa no sentido de se criar uma nova ferramenta informatizada que foi desenvolvida em Curitiba, que seria chamada de Athena⁵, para a informatização da metodologia SCoM. Esta ferramenta informatizada permitia que os colaboradores informassem à ferramenta suas competências, sendo que estas informações seriam mais tarde confirmadas pelas gerências imediatas para que pudessem efetivamente fazer parte do sistema. Hoje devido à reestruturação da área de tecnologia da informação na

⁵ O sistema Athena/Siemens é detalhado no item 2.3.2.

Siemens, o desenvolvimento e utilização desta ferramenta estão temporariamente paralisados.

O principal objetivo da metodologia SCoM (Siemens, 2006, p.04) é gerenciar as competências necessárias para a realização da atual e futura estratégia de negócios da empresa, sendo que nela, entende-se por competência, o conjunto das características que descrevem os aspectos profissionais e sociais dos atributos necessários ao cargo, compreendendo competências técnicas, em processos, sociais e pessoais, apresentados na seqüência, através da Figura 6.



Figura 6 – Competências SCoM. Fonte: (SIEMENS, 2006, p.04)

Na metodologia SCoM, conforme demonstrado na Figura 7, as competências são refletidas em conhecimentos técnicos e metodológicos, em experiência profissional e intercultural e em habilidades. Estas competências ainda estão divididas em *hard-skills*, que são conhecimentos tecnológicos, metodológicos e profissionais e os *soft-skills*, que seriam as capacidades individuais, sociais e de gestão presentes em cada colaborador. Neste sentido, observa-se que neste modelo se busca também quantificar as competências emergentes nas equipes de trabalho.



Hard skills + soft skills

Figura 7 – Elementos formadores da competência SCoM. Fonte: (SIEMENS, 2006, p.05)

Complementando a Figura 7, a Figura 8 demonstra que segundo a metodologia SCoM, as habilidades dos colaboradores são compostas por quatro indicadores de competências, que são tomar iniciativa (*drive*), desenvolver estratégias de sucesso (*focus*), conquistar parceiros (*impact*) e liderar times (*guide*)⁶.

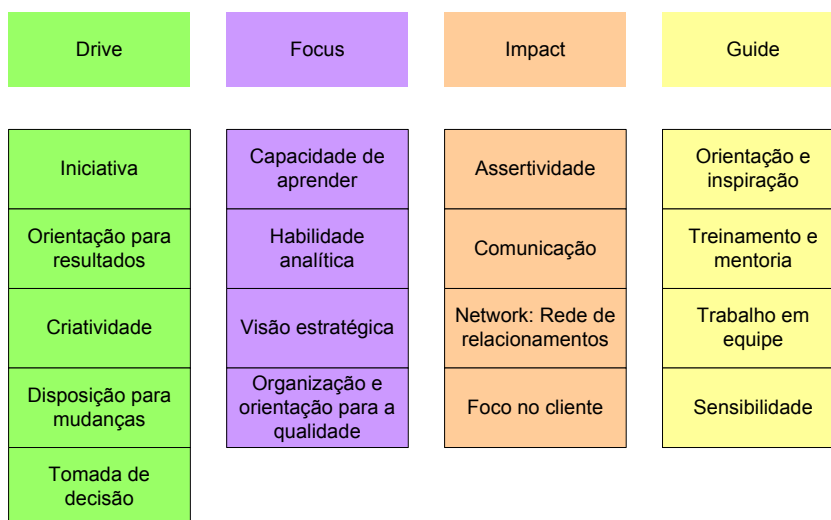


Figura 8 – Habilidades SCoM. Fonte: (SIEMENS, 2006, p.06)

O foco da metodologia SCoM na gestão por competências da Siemens é atuar no alinhamento das competências dos colaboradores da empresa, com as demandas apresentadas pelo mercado local e regional, pois, cada unidade regional da Siemens atende à um diferente nicho de mercado, constituindo diferentes células de negócio da empresa. A estratégia de negócio de cada uma destas unidades regionais da Siemens é re-alinhada a cada 5 anos, então é novamente definida a estratégia de competências (SCoM) necessária ao atendimento da estratégia.

A Figura 9 demonstra que na Siemens existem dois tratamentos distintos dados às competências, um voltado às pessoas e outro voltado aos processos. Em se tratando das pessoas, suas funções são o foco, pois elas são fatores determinantes para o bom desempenho dos colaboradores no exercício de suas funções. Quanto aos processos, a

⁶ Estão sendo utilizados os termos originais na língua inglesa devido ao fato que houve um trabalho de tradução pelo grupo Siemens. Como não é objetivo deste trabalho verificar toda a semântica das palavras, optou-se por apresentar os termos simultaneamente nos dois idiomas.

organização é o foco, pois os grandes grupos de competências derivam diretamente da estratégia de negócios das unidades.



Figura 9 – Composição das competências SCoM. Fonte: (SIEMENS, 2006, p.07)

Durante a definição da estratégia de negócios das unidades da Siemens, são definidas a visão, missão, estratégias e objetivos que aquela unidade irá seguir. Então, é definida a estratégia de competências necessária para que a empresa consiga atingir seus objetivos. Após a definição da estratégia é montado o modelo de desenvolvimento de competências que é assistido pelo sistema Athena, tratado no item 2.3.2. Nele, são analisados os recursos humanos da empresa que já estariam alinhados com a estratégia de competências, é avaliada a necessidade de treinamentos, ou seja, efetuada a identificação e gestão das competências *hard-skill* e *soft-skill*⁷. Este processo é representado pela Figura 10.



Figura 10 – Estratégia organizacional SCoM. Fonte: (SIEMENS, 2006, p.08)

⁷ Ver Figura 7.

A gestão das competências dos colaboradores (SIEMENS, 2006, p.09-19) é estruturada sobre processo demonstrado através da Figura 11. A definição das competências é realizada de acordo com o direcionamento dado à estratégia de negócios da Siemens, onde são primeiramente definidas as competências de negócio que deverão apoiar a estratégia da empresa. Em seguida são definidas as regras quantitativas e qualitativas bem como as regras de análise das competências. Logo que estabelecidas as regras, verifica-se qual é a “posição” atual das competências dos colaboradores das unidades de negócio em comparação com as necessidades futuras, que foram definidas juntamente com a estratégia da empresa. O resultado da metodologia SCoM é a diferença das competências que a empresa possui com as competências que serão necessárias futuramente. Após a análise deste resultado é estruturado um plano de ação, contendo as necessidades de competências para atender a nova estratégia de negócio, observando sua relevância para o negócio e para as tendências do mercado.

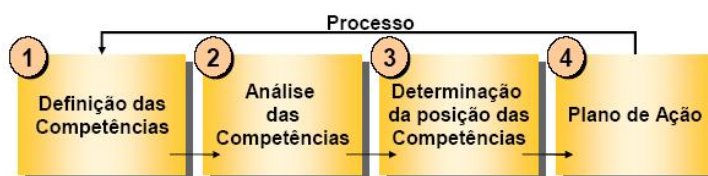


Figura 11 – Processo SCoM. Fonte (SIEMENS, 2006, p.9-19)

Este processo demonstrado acima através da Figura 11, norteia o funcionamento da ferramenta Athena, que será discutida no item 2.3.2.

2.1.4.3 Metodologia de avaliação de competências *Efix Competências*

A metodologia utilizada pelo *Sistema Efix Competências* para avaliação de competências é chamada de “metodologia de avaliação em 360 graus”. Segundo (Efix – Visão Geral, 2005 p. 7-10), recebe esta denominação pelo fato de que durante um processo de avaliação, podem ser adicionadas questões para a avaliação de indicadores de competências de acordo com o modelo buscado. Para Reis (2000, p.10-12), o processo de avaliação em 360 graus é uma ferramenta onde avaliadores e avaliados geram *feedbacks* multidirecionais de forma simultânea. Seiffert (2002, p. 151) complementa que a metodologia de avaliação de competências 360 graus é uma das formas mais confiáveis de se avaliar elementos intangíveis. Através dela, o colaborador é avaliado por seus clientes internos, colegas de trabalho, fornecedores internos, ele próprio e sua chefia imediata, dentre outros.

[...] a avaliação 360 graus possibilita inputs que permitem identificar pontos fortes e oportunidades de melhorias individuais, a partir de percepções de vários ângulos: pares, subordinados, clientes e fornecedores internos. E mais do que isto possibilita a reflexão entre suas próprias percepções e de outras pessoas envolvidas no processo, podendo ter como consequência uma modificação de atitudes e comportamentos dentro e fora da organização Reis (2000) apud (UBEDA, 2002, p.07)

Segundo a própria empresa criadora do software *efix competências* (Efix – Visão Geral, 2005 p. 5), busca-se através da análise qualitativa do colaborador, mensurar de forma mais precisa as competências de seus colaboradores. Para isso, a análise é efetuada pela aplicação de questionários eletrônicos definidos por cargo e distribuídos inicialmente de forma cruzada entre subordinado e chefe. Porém esta distribuição cruzada de avaliações pode ser estendida para colegas de trabalho, chefias, enfim, todas as pessoas que de alguma forma interajam com o avaliado, criando a chamada avaliação em 360 graus. Nela, para cada cargo da corporação é definido um perfil esperado com valores mínimo e máximo para cada competência.

A Figura 12 demonstra através da metodologia 360 graus, pode-se criar em um cenário com vários questionários de avaliação de competências e ligá-los com diversos colaboradores em posições distintas. Por exemplo, um diretor durante o processo de avaliação de um subordinado, poderia utilizar um questionário diferente do que um gerente utilizaria para avaliar outro gerente. Esta avaliação de competências pode também ser construída a partir de uma posição mínima e máxima esperada para cada competência, para posterior comparativo de valores através de escalas pré-definidas. As gerências têm a possibilidade de criar diferentes processos de avaliação na forma de *workflow* para cada posto de trabalho, definindo até mesmo quem será avaliado primeiro e quem poderá sugerir e quais competências adicionais no processo.

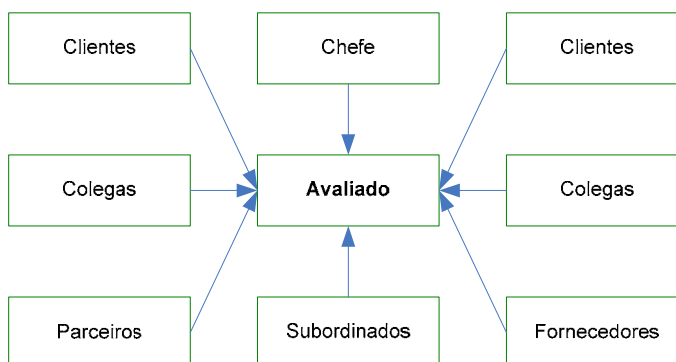


Figura 12 – Avaliação 360 graus. Fonte: Efix Avaliações (2005, p. 07)

Se a pretensão é de avaliar competências, e isto terá influência na carreira ou remuneração do indivíduo o recomendado é utilizar a avaliação 360 graus, pois ela permite mensurar o nível de atendimento das competências desejadas. Como a avaliação é subjetiva, que prevaleça a opinião de vários pontos de vista ao invés de somente a da chefia imediata Edwards (1996) apud (SEIFFERT, 2002, p. 151; 205-207).

2.1.5 Considerações sobre competências

Conclui-se através desta análise de diversos autores comportamentalistas e construtivistas em um contexto onde há interdisciplinaridade, multidisciplinaridade, etc. a necessidade do trabalho em equipe existe, onde têm-se as dimensões de relacionamento social, profissional (ou técnica) e de negócio.

Como a competência é uma característica essencialmente social, se um indivíduo tem algo muito bom, mas não teve a competência de articular seus conhecimentos juntamente com a base social, há uma grande possibilidade de que esta competência não seja manifestada. O que não impede que, caso este esteja imerso em um determinado contexto no futuro que permita sua articulação com a base social, esta competência apresente-se como uma propriedade emergente, sendo a competência uma emergência de conhecimentos, habilidades e atitudes.

Porém, existem muitos conhecimentos que são adquiridos no ambiente escolar que não são articuláveis, mas aparentemente, existem aspectos que seriam demonstráveis. A pessoa poderia responder a muita coisa, mas não saber o que está fazendo. Destas pessoas, as que têm atitudes, capacidade de articular o conhecimento aparente, do volume já restringe o conhecimento articulável, e, dos que tem vontade realmente de tomar uma ação, de provocar, estão em um conjunto ainda menor. Não querendo dizer que a competência é uma dimensão menor, mas mapeada sobre o conhecimento demonstrado ela é menor.

Também poderão existir competências que só serão articuladas quando existe um determinado nível de confiança entre as partes, evidenciando a existência de uma trajetória histórica. No contexto deste trabalho, a competência objetivada não será baseada em

trajetória histórica e relacionamentos, mas sim a competência que está na capacidade de articular ações, explicitada através de indicadores curriculares coletados através de ferramentas informatizadas.

Durante esta coleta de informações que podem evidenciar a existência de determinadas competências dos indivíduos, é possível que a ferramenta informatizada depare-se com situações onde exista um volume de informações que aponte quantitativamente para a existência de um indicador de competência em potencial. Porém analisando-se qualitativamente este indício, existe a probabilidade de que se chegue à conclusão de que ele não represente a relevância que aparenta. Ou seja, o volume indicado até pode ser um bom indício, apresentando um esforço do indivíduo perante um cenário de aceitação social ao invés da manifestação de um conjunto de competências. Esta aceitação social pode ser desencadeada por um simples modismo ou outro fator, estando fora do escopo deste trabalho.

2.1.5.1 Tipologia para busca de indicadores de competências

De acordo com as metodologias discutidas como a SCoM desenvolvida pela Siemens e a utilizada pelo sistema Efix, observa-se que a categorização de conhecimentos, habilidades e atitudes direciona as análises às competências profissionais. As competências em processos não podem neste momento serem analisadas por possuírem características atemporais. As competências sociais também estão fora do escopo deste trabalho, pois verificam a forma através da qual as pessoas se relacionam no ambiente de trabalho. Da mesma forma, as competências pessoais são elementos tácitos da pessoa. Ou seja, diante do conjunto de conhecimentos, habilidades e atitudes que compõe o universo das competências que delineiam o perfil dos indivíduos, é possível observar com base na teoria discutida anteriormente, que as competências sociais e pessoais são elementos tácitos íntimos do ser. Observando-se sob a mesma ótica, as competências técnicas, segundo demonstra a metodologia SCoM⁸ são periféricas⁹ em relação aos demais tipos de competências e são fundamentadas em conhecimentos sistematizados e tangíveis.

⁸ Vide 2.1.4.2.

⁹ Conforme demonstra a Figura 6.

Busca-se então, através da observação das competências técnicas dos indivíduos, organizar uma tipologia que tenha a possibilidade de demonstrar indícios das capacidades da pessoa. Neste sentido, conclui-se com base na teoria já discutida que, uma vez que as competências técnicas são passíveis de mensuração, fazem parte deste estudo as competências adquiridas no ambiente acadêmico oriundo de sua formação. Isso permite que o sistema informatizado que fará a busca por indícios de competências em uma base curricular, tenha a possibilidade de auxiliar o gestor a articular de uma melhor forma, a convergência dos egressos para a disciplina de engenharia de produção.

Esta busca será efetuada através da análise de variáveis observadas nos currículos dos egressos, que podem sugerir informações tais como graduação, formação complementar, trabalhos apresentados, participação em projetos, áreas de atuação e toda a informação contida no currículo. Estas informações além de fazerem parte das competências técnicas do indivíduo, são passíveis de verificação nos arquivos padrão XML gerados pelo sistema de Currículos Lattes¹⁰ do CNPq.

São estes arquivos, gerados pelo sistema de Currículos Lattes que compõe a base curricular que irá alimentar o sistema informatizado. Esta ferramenta informatizada será desenvolvida com base nas tecnologias analisadas no item 2.2, explorando as metodologias apresentadas no item 2.3, e terá seu funcionamento demonstrado no Capítulo 3.

2.2 LINGUAGENS DE PROGRAMAÇÃO E BASES DE DADOS UTILIZADOS NA CONSTRUÇÃO DE SISTEMAS

Nesta seção será efetuado um estudo das linguagens de programação e bases de dados existentes hoje no mercado, buscando verificar quais são as opções mais relevantes a serem utilizadas na construção do referido sistema.

Busca-se através do estudo das atuais tecnologias em linguagens de programação e bases de dados, obter um referencial teórico que permita analisar quais as melhores alternativas para aplicação no projeto, além de fornecer material que permita efetuar a análise técnica dos sistemas existentes para a gestão de competências. Serão analisados diversos detalhes tais como características das linguagens, se elas são ou não flexíveis, se

¹⁰ O Sistema de Currículos Lattes é discutido no capítulo 2.3.3.

são livres ou proprietárias, qual é o modelo de base de dados mais indicado, devido às características necessárias ao projeto dentre outras particularidades.

Ao final desta seção, obter-se-á um quadro conceitual de deverá orientar na seleção da linguagem de programação e base de dados que serão utilizados na construção do protótipo que no Capítulo 3, efetuará a simulação com base na teoria discutida neste capítulo.

2.2.1 Conceitos de linguagens de programação

A origem das tecnologias existentes utilizadas para o desenvolvimento de sistemas informatizados remontam o início da era da computação, onde a programação de sistemas era executada em dispositivos de *hardware*, pela ligação de fios entre pontos de placas de circuito impresso para a geração de interrupções.

A primeira linguagem de desenvolvimento de programas de computador foi o Fortran, criado em 1954. Com o passar dos anos e a evolução das linguagens de programação, surgiu uma linguagem de desenvolvimento que ainda está em uso em sua forma original na Europa, chamada de BCPL e desenvolvida por Martin Richards. Esta linguagem anos mais tarde evoluiria para dar origem à linguagem de desenvolvimento de sistemas conhecida como “Linguagem B” que foi produzida por Ken Thomson. Esta por sua vez, influenciaria no início da década de 1970 o desenvolvimento da “Linguagem C”, largamente utilizada e considerada a mãe de todas as linguagens modernas (SCHILDT, 1997, p. 01-03) (HOLZNER, 2001, p.01-04).

Segundo Santos Neto (2001, p. 83), através dos esforços dos cientistas Wirth Hoare e Jacopini, foram lançados acerca de 1966 os fundamentos que dariam origem à programação estruturada a partir da evolução das técnicas de desenvolvimento de sistemas imperativas e funcionais (Sebesta, 2000, p. 38), permitindo que as técnicas de controle de interrupções passassem a ser executadas através de software.

Desde então, as linguagens de construção de sistemas passaram por paradigmas distintos em sua evolução, onde se popularizou através do paradigma do *desenvolvimento estruturado*, sendo que hoje se vivencia de forma plena a era do paradigma da *orientação à objetos*. Sua linha de sucessão aponta em um futuro próximo para o *paradigma dos sistemas agentes*.

Linguagens de programação estruturadas

A programação estruturada foi uma das primeiras técnicas para o desenvolvimento de ferramentas informatizadas a surgir e difundir-se no mercado, substituindo as técnicas conhecidas como “técnicas funcionais”¹¹ Yourdon (1988, p. 30). Contribuiu em grande monta para a disseminação dos computadores pessoais, tidos como os principais responsáveis pela disseminação e barateamento da tecnologia presente nos dias de hoje.

Esta modalidade de construção de programas de computador estabelece um conjunto de regras de desenvolvimento baseada em algoritmos¹². Também conhecido como pseudo-linguagem, o algoritmo é uma forma generalista de se descrever um programa estruturado de computador que organiza procedimentos bem definidos, onde suas etapas de execução são escritas na linguagem¹³ de quem o descreve, o que possibilita o seu entendimento por qualquer pessoa, conforme demonstra a Figura 13.

```
Inicio Programa_01 ()
Se (condição = Verdadeiro) faça:
    1. Execute instrução  $\alpha$ ;
    2. Execute instrução  $\beta$ ;
Senão faça:
    1. Execute instrução  $\Omega$ ;
Fim do Programa_01 ();
```

Figura 13 – Exemplo de Pseudocódigo

Já o fluxograma segundo Ascencio et al (2002, p. 4-5) é um mecanismo que permite transcrever o pseudocódigo em formato gráfico, apresentando o programa na forma de um ou vários processos e sub-processos, denotando o mecanismo através do qual

¹¹ Como exemplo de linguagem desta técnica, cita-se o Fortran.

¹² Segundo o dicionário Aurélio, algoritmo é um “Conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema, ou de uma classe de problemas, em um número finito de etapas”. Seguindo a mesma linha de raciocínio, pode-se citar os autores (LOUDON, 2000, p. 05) e (CORMEN et al., 1998, p.01).

¹³ Neste caso, entende-se linguagem como idioma do indivíduo, sendo que o mesmo pode transcrever seus programas de computador em pseudocódigo em qualquer outro idioma.

ocorre o fluxo de suas informações e operações, ilustrados esquematicamente na Figura abaixo.

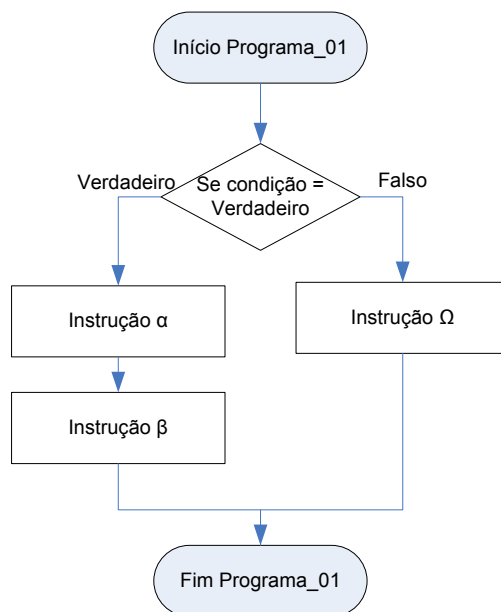


Figura 14 – Exemplo de Fluxograma

Desta forma, na programação estruturada demonstra-se através de algoritmos ou fluxogramas o conjunto de mecanismos necessários para controlar a execução de um programa de computador através das funcionalidades existentes na linguagem adotada.

Seu princípio de funcionamento é de que o software é composto por blocos de código, que interagem através de mecanismos com início e término da execução de instruções bem definidos. O Quadro 1 demonstra as características de algumas das principais linguagens de programação presentes ainda hoje no mercado, que seguem o paradigma da programação estruturada.

Quadro 1 – Linguagens de programação do paradigma estruturado

Nome	Código nativo	Código interpretado	Fortemente tipada	Fracamente tipada	Plataforma windows	Plataforma linux	Difusão na academia (PR)	Difusão no mercado (PR)
Clipper	X		X					
Pascal	X		X		X	X	X	
C	X			X	X	X	X	X
Lisp		X		X	X	X		

Fonte: o autor

No contexto atual, através de análise dos currículos dos cursos superiores de tecnologia das instituições de ensino renomadas da região de Curitiba, foi verificado que nenhuma delas possui mais cursos de Clipper ou Lisp; apenas a UFPR e PUC oferecem disciplinas de Inteligência Artificial, abordando uma variante simplificada do Lisp, o Prolog. Em todas, verificou-se que os alunos têm conhecimento nas linguagens C e Pascal nos primeiros anos do curso, onde estes conhecimentos adquiridos servirão como base, para posterior compreensão das linguagens de programação orientadas à objetos C++ e Delphi, tratadas no decorrer deste trabalho.

O emprego das técnicas de desenvolvimento estruturadas internamente em outras linguagens e ambientes que permitem a construção de programas para recentes plataformas computacionais, sendo empregadas em todo o dispositivo computacional existente, desde computadores de grande porte até telefones celulares.

Existe uma categoria de sistemas informatizados que utiliza principalmente técnicas estruturadas de desenvolvimento em sua construção, que são os “sistemas embarcados¹⁴”. Esta categoria de sistemas reside em pequenos *hardwares* como catracas

¹⁴ Os sistemas embarcados são comumente referenciados na literatura por seu termo original em inglês “*embedded systems*”.

eletrônicas, sistemas de injeção eletrônica de combustível de automóveis, centrais telefônicas, fornos de micro ondas, dentre outros aparatos tão comuns no dia a dia.

Esta geração de linguagens de desenvolvimento que utiliza diferentes técnicas e procedimentos, objetiva tornar o processo de desenvolvimento mais natural, sendo conhecida como “Orientação à Objetos”.

Linguagens de programação orientadas à objetos

Barbalho (2002, p. 04) demonstra que, historicamente os fundamentos da orientação para objetos tiveram sua origem no início da década de 60, com o desenvolvimento de uma linguagem de simulação chamada de *Simula I*, a qual tinha como objetivo permitir além da simulação de sistemas, suportar a descrição dos mesmos. Porém, somente anos mais tarde é que foi desenvolvido o primeiro sistema com a tecnologia de objetos, que foi o projeto *Dinabook* da Xerox, que mais tarde resultou na linguagem de programação *Smalltalk*.

Desde então, a metodologia de construção de programas de computador orientada para objetos, ganhou espaço sobre as metodologias tradicionais por apresentar maior naturalidade conceitual, pelo fato de possibilitar que entidades¹⁵ pudessem ser representadas da forma que elas realmente são, apresentando-se como a grande tendência da engenharia de software da atualidade.

Assim esta metodologia, busca identificar o conjunto de sujeitos, ou “objetos” que melhor representa o programa de computador desejado. O funcionamento do programa ocorre de acordo interação de eventos entre os objetos componentes do sistema que podem ser herdados de outras classes de objetos. As palavras de (DATE, 2000, p. 702), fundamentam que os objetos são abstrações de dados do mundo real e possuem estruturas de dados e de funcionalidades que o objeto representado possui.

No intuito de facilitar o entendimento, pode-se utilizar como exemplo um objeto ventilador, onde este possui determinado arcabouço estrutural o qual permite ao indivíduo

¹⁵ Entidades são abstrações do mundo real ou virtual, doravante denominados objetos de um sistema. Estes objetos poderão assumir o papel de quaisquer sujeitos como automóveis, pessoas, equipamentos, documentos, etc.

operá-lo. Para operar este objeto ventilador, a pessoa interage através de um conjunto de botões que o ventilador herdou de outra classe de botões, que pode figurar como um “seletor de velocidades”, o qual dispara eventos que efetuam as mudanças de estado no objeto ventilador, iniciando seu funcionamento, alterando sua velocidade de rotação ou desligando-o. Neste sentido, segundo (DATE, 2000, p. 698), um objeto é descrito através de seus atributos, que podem ser acessados ou modificados através das operações definidas no próprio objeto, sendo que estas operações são chamadas de métodos. Seguindo o mesmo exemplo, seus atributos seriam suas velocidades de rotação, cor, acabamento, tamanho, voltagem de alimentação, número de pás, etc. e seus métodos seriam aumentar e reduzir a velocidade, ligar, desligar, etc. que seriam acionados por um segundo objeto denominado seletor de velocidades.

Esta possibilidade de representar entidades do mundo real na forma de objetos no programa computacional, de forma que cada objeto possui estruturas que o permitem executar ações conforme solicitação dele próprio ou de outros objetos, auxilia a manter o código do sistema organizado, padronizado, devido aos conceitos que a própria metodologia fornece (Barbalho, 2002, p. 04).

A finalidade do uso de técnicas de desenvolvimento de programas de computador orientados para objetos, é de complementar os mecanismos existentes na programação estruturada e não substituí-la. Pode-se assumir que a programação orientada para objetos é uma evolução do paradigma de desenvolvimento de software estruturado, onde são implementados mecanismos que objetivam facilitar o processo de construção de software.

Além disso, o paradigma de objetos permite que sejam desenvolvidos pacotes que são chamados de “bibliotecas de classes”, que facilitam a reutilização de código através de mecanismos de herança em diferentes programas de computador. Uma das grandes vantagens do uso de classes é na organização e manutenção do código escrito, pois quando o código-fonte é alterado em uma classe, todas as demais partes do aplicativo que “consomem” desta funcionalidade automaticamente passam a utilizar o código atualizado após o processo de compilação.

O Quadro 2 demonstra as características de algumas das principais linguagens de programação presentes no mercado, que implementam a programação orientada a objetos.

Quadro 2 – Linguagens de programação do paradigma orientado à objetos

Nome	Possui IDE	Código nativo	Código intermediário	Código interpretado	Fortemente tipada	Fracamente tipada	Suporte a threads	Plataforma windows	Plataforma linux	Plataforma web	Livre	Proprietária	Difusão na academia (PR)	Difusão no mercado (PR)
Delphi	X	X			X		X	X				X	X	X
Visual Basic .NET	X		X		X		X	X	X	X		X	X	X
C++	X	X			X		X	X	X		X		X	X
C# .NET	X		X		X		X	X	X	X				X
Java	X		X		X		X	X	X	X			X	X
PHP	X			X		X		X	X	X	X		X	X
CLOS	X			X	X		X	X	X		X			

Fonte: o autor

No contexto atual, através de análise dos currículos dos cursos superiores de tecnologia das instituições de ensino renomadas da região de Curitiba, foi verificado que nenhuma delas possui disciplina que discuta os mecanismos de implementação da linguagem orientada a objetos derivada do Lisp, o *Common Lisp Object System* CLOS. Das linguagens orientadas a objetos, verificou-se que os alunos têm conhecimento nas linguagens C++, Visual Basic .NET, Delphi, Java e PHP.

Também não foi encontrada em nenhuma das instituições de ensino verificadas, disciplina de linguagem de programação que contemple o estudo do C# .NET. Porém, como esta linguagem é bastante semelhante ao Java e ao C++, entende-se que, quem tem conhecimento nestas duas linguagens, têm a possibilidade de trabalhar com o C# .NET após um período de treinamento com seu ambiente de desenvolvimento de software.

2.2.2 Conceitos de bases de dados

Bases de dados são sistemas que possuem estruturas que possibilitam o armazenamento e recuperação de dados (HEUSER, 2001, p.05). Neles, os dados geralmente estão organizados de forma hierárquica, onde o Banco de Dados (que pode atuar como sistema gerenciador) tem o nível mais alto. Nesta hierarquia os campos formam

um registro e os registros formam uma tabela. A entidade de um Banco de Dados é uma superclasse onde registros específicos têm seus respectivos atributos, ou campos, ou itens de dados (LAUDON, 1999, p. 126).

Para Cheu (1990, p. 1-6) o conjunto de registros organizados de forma matricial em linhas e colunas é conhecido como tabela de dados, sendo que cada coluna que representa o fragmento de um registro é conhecido como campo. O conjunto de campos que compõe uma linha de uma tabela é chamado de registro da tabela. Date (2000, p. 54) afirma que uma *tupla* é um conjunto de registros.

O armazenamento físico dos dados em um banco de dados é executado de forma não ordenada, ou seja, imagina-se um banco de dados como um livro com as páginas desordenadas, onde, o índice indica onde está a informação que se está buscando. Caso não exista índice, é necessário percorrer toda a tabela até que o dado desejado seja encontrado. Este procedimento de busca não indexada é conhecido por busca seqüencial.

Existem basicamente dois tipos de índices, segundo Korth (1995, p. 270-276):

- índice ou chave primária: indica que o campo do registro é único na tabela de dados;
- índice ou chave secundária ou ainda chave estrangeira: é um campo indexado em um registro de uma tabela de dados, mas esta modalidade de índice não identifica este campo como único, ou seja, podem existir infinitos outros registros idênticos.

A evolução da abordagem tradicional de Banco de Dados é o conhecido Sistema Gerenciador de Banco de Dados Relacional. Esta abordagem é amplamente utilizada, organizando seus dados através de relações, além de permitir o uso simultâneo dos dados por muitos sistemas aplicativos, que podem operar de forma independente e concorrente.

Sistema Gerenciador de Banco de Dados

O Sistema Gerenciador de Banco de Dados¹⁶ (SGBD) é um software onde os dados armazenados por sistemas aplicativos podem ser manipulados através de uma

¹⁶ Tradução do termo na língua inglesa *Database Management System* (DBMS).

interface de acesso, garantindo confiabilidade e integridade nos processos de armazenamento e recuperação dos dados através de mecanismos chamados de transação. No mercado, os sistemas gerenciadores de grande porte mais conhecidos são os produtos Oracle, SQL Server da Microsoft, DB2 da IBM e o software livre PostgreSQL.

Já no segmento dos sistemas gerenciadores de banco de dados de pequeno porte destacam-se os produtos comerciais Access e FoxPro da Microsoft, sendo que o segundo é integrado à linguagem de programação de mesmo nome, o Interbase da Borland e o MySQL da AB Software. Neste mesmo segmento existem os softwares livres Firebird criado a partir do já citado Interbase e a versão gratuita do também citado MySQL.

Laudon (1999, p. 129) e Heuser (2001, p.78) fundamentam que o modelo relacional foi desenvolvido para superar as limitações de outros modelos na representação e relacionamento de dados, representando seus dados através de tabelas que são combinadas através de relações.

O SGBD possui quatro características básicas que definem sua estrutura segundo Korth (1995, p. 13-14):

- Independência dos dados, que possibilita alterar a estrutura dos dados em um nível sem afetar os outros.
- Linguagem de definição de dados, que exprime a forma através da qual cada elemento deve ser representado no banco de dados.
- Linguagem de manipulação de dados, que é uma linguagem que possibilita o manuseio dos dados no sistema gerenciador. A mais difundida hoje é o SQL (*Structured Query Language*), que está presente em todos os sistemas gerenciadores de bancos de dados existentes no mercado. Todavia, cada fabricante de sistema gerenciador de banco de dados também implementa sua própria linguagem além do SQL, como por exemplo no caso do Oracle, existe o PL/SQL, o PostgreSQL possui o PL/PGSQL, etc.
- Dicionário de dados, que é a funcionalidade que permite o armazenamento das definições dos elementos de dados existente no banco de dados.

Bases de dados orientadas à objetos

A filosofia de representação e manipulação de dados conhecida mundialmente por OOP (*Object Oriented Programming*) ou programação orientada à objetos já tratada

anteriormente no item 2.2.1 deste trabalho, deu origem ao desenvolvimento de sistemas gerenciadores de bancos de dados que seguissem a mesma filosofia.

Dos sistemas gerenciadores de bancos de dados disponíveis no mercado, os bancos de dados que suportam o paradigma da orientação a objetos mais conhecidos são o recém-descontinuado Jasmine, que foi o primeiro banco de dados que suportou exclusivamente o modelo orientado a objetos, foi produzido pela Visionnaire e o já mencionado software livre PostgreSQL, que é um banco de dados híbrido do tipo objeto-relacional.

Para Laudon (1999, p. 135) o banco de dados orientado à objetos permite o armazenamento dos dados na forma de objetos, tal qual eles apresentam-se no mundo real. Neles (nos objetos), estão incluídas as instruções de processamento para completar cada uma de suas transações, visando facilitar o desenvolvimento de software por meio da reutilização de código e capacidade de construir novas aplicações que combinam vários tipos de informação, como a multimídia e aplicações web, por exemplo.

A aproximação entre a linguagem de desenvolvimento orientada em objetos e o sistema gerenciador de banco de dados ainda hoje é um problema, pois, para armazenar os dados de um sistema construído totalmente orientado a objetos em um banco relacional, faz-se necessário o uso de uma técnica chamada de “camada de persistência”. Esta camada de software, informa aos objetos responsáveis pela gravação e recuperação das informações para onde os dados devem ir. Isto pode ocasionar lentidão no sistema, além de dificultar bastante o processo de desenvolvimento pela necessidade de implantar uma “camada de software” entre a aplicação e o sistema gerenciador de banco de dados.

Bases de dados XML

O XML, abreviação para *Extensible Markup Language* é uma especificação originalmente desenvolvida pelo W3C (*World Wide Web Consortium*) com o intuito de propiciar o intercâmbio de informações, como sucessora do padrão SGML (*Standard Generalized Markup Language*). Tornou-se também um importante elemento para viabilizar o intercâmbio de informações entre diferentes sistemas de informação existentes na rede Internet (W3C, 2006).

O padrão HTML proporciona que *tags*¹⁷ sejam utilizadas para definir padrões para apresentação de dados, como formatação de parágrafos, por exemplo. Já o padrão XML conforme demonstra a Figura 15, fornece um formato pré-definido onde se pode armazenar estruturas de dados através destas marcas, facilitando a manipulação de dados em múltiplas plataformas e ambientes de sistemas informatizados.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <CURRICULO-VITAE SISTEMA-ORIGEM-XML="LATTES_OFFLINE" DATA-ATUALIZACAO="06022007" HORA-ATUALIZACAO="142131"
3 xmlns:lattes="http://www.cnpq.br/2001/XSL/Lattes"><DADOS-GERAIS NOME-COMPLETO="Dani Juliano Czelusniak"
4 NOME-EM-CITACOES-BIBLIOGRAFICAS="CZELUSNIAK, D. J." NACIONALIDADE="B" CPF="02440056995" PAIS-DE-NASCIMENTO="Brasil"
5 UF-NASCIMENTO="PR" CIDADE-NASCIMENTO="Campo Largo" DATA-NASCIMENTO="27071976" SEXO="MASCULINO"
6 NUMERO-IDENTIDADE="603330468" ORGAO-EMISSOR="IP" UF-ORGAO-EMISSOR="PR" DATA-DE-EMISSAO="02081990"
7 NUMERO-DO-PASSAPORTE="" NOME-DO-PAI="José Carlos Czelusniak" NOME-DA-MAE="Sonia Maria Czelusniak"
8 PERMISSAO-DE-DIVULGACAO="NAO" OUTRAS-INFORMACOES-RELEVANTES="Framework Microsoft .NET e Mono Banco de Dados Relacional e
9 Objeto Relacional Unified Modeling Language (UML) Programação Orientada a Objetos (OOP) Programação Orientada a Agentes (AOP) JADE -
10 Java Agent Development Framework com Visual Studio e C# (10/10/2005)"><ENDERECO
11 FLAG-DE-PREFERENCIA="ENDERECO_INSTITUCIONAL"><ENDERECO-PROFISSIONAL CODIGO-INSTITUICAO-EMPRESA="001800000992"
12 NOME-INSTITUICAO-EMPRESA="Faculdade Cenecista de Campo Largo" CODIGO-UNIDADE="" NOME-UNIDADE="" CODIGO-ORGAO=""
13 NOME-ORGAO="" PAIS="Brasil" UF="PR" LOGRADOURO-COMPLEMENTO="Rua Rui Barbosa, 541" BAIRRO="Centro" CIDADE="Campo Largo"
14 CAIXA-POSTAL="" CEP="83601140" DDD="41" TELEFONE="33922211" RAMAL="" FAX="33922211" E-MAIL="dani.juliano@gmail.com"
15 HOME-PAGE="http://www.presidentekennedy.br/"><ENDERECO-RESIDENCIAL PAIS="Brasil" UF="PR" LOGRADOURO="" BAIRRO=""
16 CIDADE="Campo Largo" CAIXA-POSTAL="" CEP="" DDD="" TELEFONE="" RAMAL="" FAX="" E-MAIL=""
17 HOME-PAGE="http://www.presidentekennedy.br/~dani.juliano/"><ENDERECO><FORMACAO-ACADEMICA-TITULACAO><GRADUACAO
18 SEQUENCIA-FORMACAO="1" NIVEL="1" TITULO-DO-TRABALHO-DE-CONCLUSAO-DE-CURSO="" NOME-DO-ORIENTADOR=""
19 CODIGO-INSTITUICAO="000200000993" NOME-INSTITUICAO="Faculdade de Ciência e Tecnologia do Paraná" CODIGO-CURSO="90000001"
20 NOME-CURSO="Tecnologia Em Processamento de Dados" CODIGO-AREA-CURSO="10300007" STATUS-DO-CURSO="CONCLUIDO"
21 ANO-DE-INICIO="1993" ANO-DE-CONCLUSAO="1997" FLAG-BOLSA="NAO" CODIGO-AGENCIA-FINANCIADORA=""
22 NOME-AGENCIA=""><ESPECIALIZACAO SEQUENCIA-FORMACAO="2" NIVEL="2" TITULO-DA-MONOGRAFIA="" NOME-DO-ORIENTADOR=""
23 CODIGO-INSTITUICAO="000300000995" NOME-INSTITUICAO="Faculdades Spei" CODIGO-CURSO="90000029" NOME-CURSO="Bancos de Dados"
24 STATUS-DO-CURSO="CONCLUIDO" ANO-DE-INICIO="1998" ANO-DE-CONCLUSAO="1998" FLAG-BOLSA="NAO"
25 CODIGO-AGENCIA-FINANCIADORA="" NOME-AGENCIA="" CARGA-HORARIA="360"><ESPECIALIZACAO SEQUENCIA-FORMACAO="3" NIVEL="2"
26 TITULO-DA-MONOGRAFIA="Utilização do Sistema Operacional Linux em Aplicações Tempo Real" NOME-DO-ORIENTADOR="Douglas Renaux"
27 CODIGO-INSTITUICAO="198100000000" NOME-INSTITUICAO="Centro Federal de Educação Tecnológica do Paraná" CODIGO-CURSO="90000003"
28 NOME-CURSO="Métodos para Engenharia de Software" STATUS-DO-CURSO="CONCLUIDO" ANO-DE-INICIO="2001" ANO-DE-CONCLUSAO="2002"
29 FLAG-BOLSA="NAO" CODIGO-AGENCIA-FINANCIADORA="" NOME-AGENCIA="" CARGA-HORARIA="360"><ESPECIALIZACAO

```

Figura 15 – Arquivo XML visualizado em modo "Código Fonte"

Com a crescente incorporação do padrão XML em diferentes sistemas informatizados, sejam eles gerenciadores de banco de dados até editores de texto, ele é definido hoje como formato padrão para estruturação e intercâmbio de informações, tanto na rede Internet quanto em sistemas locais.

¹⁷ *tag* ou marca, é uma estrutura utilizada em linguagem de marcação, que são conjuntos de códigos aplicados à textos e dados, onde dados e formatações são definidos através de instruções com demarcação de início e fim. Exemplo: <negrito> Texto </negrito>.

A Figura 16 apresenta internamente um arquivo XML, visualizado na forma de blocos de dados com o intuito de facilitar a compreensão de seu conteúdo, onde se pode perceber que os dados estão estruturados na forma blocos de textos, tabelas, imagens, etc., obedecendo às regras definidas pelo W3C, o que permite que estes arquivos de dados sejam compreensíveis aos outros sistemas.

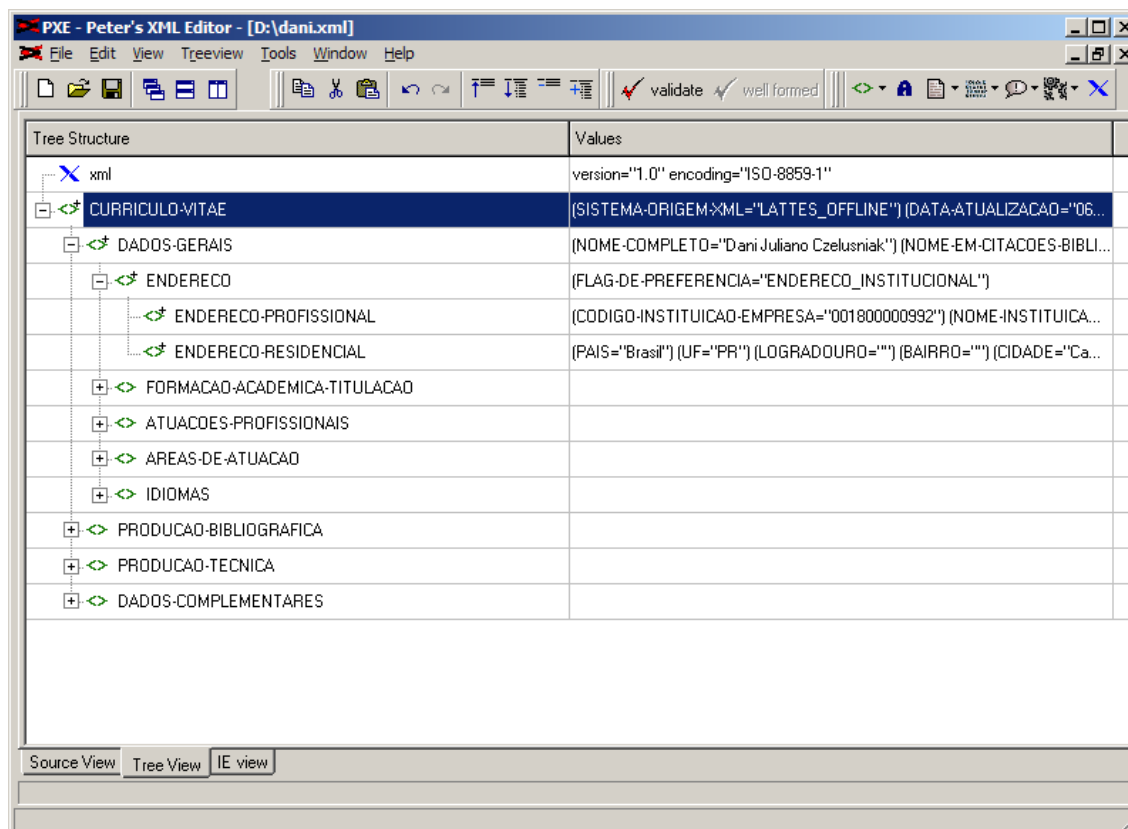


Figura 16 – Arquivo XML visualizado em modo “Blocos de Dados”

Esta padronização estrutural das *tags* permite que sejam executadas buscas em vários bancos de dados sem que necessariamente conheça-se a sua estrutura interna, permitindo que dados possam ser facilmente localizados.

Neste sentido, o XML é um formato largamente utilizado, pois é simplesmente uma notação em formato texto hierarquicamente estruturada e, apresenta a característica de produzir como resultado final arquivos de dados que podem ser utilizados em qualquer aplicação.

Um arquivo em formato XML pode ser ligado a uma especificação conhecida como DTD (*Document Type Definition*), que descreve os componentes de um documento

juntamente com todas as suas possíveis relações. O DTD funciona como um dicionário de dados de um banco de dados XML. Esta facilidade permite que sejam descritos de forma mais clara os conceitos e relações para a estruturação de conceitos e relações, permitindo que suas *tags* possam armazenar dados, e, conseqüentemente, permitindo também que possam ser estruturados bancos de dados complexos a partir de suas estruturas.

Neste sentido, o Quadro 3 apresentado abaixo demonstra as principais características dos principais produtos em bancos de dados existentes hoje no mercado e sua difusão no mercado e academia.

Quadro 3 – Bancos de dados e suas características

Nome do Produto	Relacional	Orientada a objetos	Suporte à replicação	Linguagem SQL	Linguagem própria	XML	Plataforma Windows	Plataforma Linux	Livre	Proprietário	Difusão no mercado (PR)	Difusão na academia (PR)
Oracle	X		X	X	X	X	X	X		X	X	X
SQL Server	X		X	X	X	X	X			X	X	X
PostgreSQL	X	X	X	X	X	X	X	X	X		X	X
Jasmine		X		X			X			X		
Interbase	X		X	X			X			X	X	X
Firebird	X		X	X			X	X	X		X	X
DB2	X		X	X	X	X				X	X	
MySQL	X			X	X		X	X	X	X	X	X
Visual Fox Pro	X			X	X	X	X			X		
Access	X			X		X	X			X	X	

Fonte: o autor

2.2.3 Considerações sobre linguagens de programação e bases de dados

Observando-se o escopo do projeto, que é o desenvolvimento de uma ferramenta informatizada para auxílio ao gestor, efetuar de uma melhor forma a gestão de competências, foi efetuada esta análise das principais ferramentas de linguagem de programação e bancos de dados presentes hoje no mercado, com o intuito de se verificar qual a melhor opção a ser utilizadas na construção do referido sistema.

De acordo com o estudo efetuado sobre as linguagens de programação, foi verificado que as linguagens que atendiam ao padrão estruturado de desenvolvimento gradativamente retiraram-se do mercado. Isto torna proibitivo o uso de uma linguagem estruturada tal qual o Pascal ou Clipper, devido ao fato de que estas tecnologias tornaram-se legado (exceto em sistemas especialistas e embarcados), o mercado não permitiria uma sobrevida ao sistema a ser desenvolvido. O ambiente Lisp seria uma possibilidade, por ser uma linguagem de programação bastante difundida na comunidade da Inteligência Artificial pela facilidade de implementação de modelos matemáticos, porém não são encontrados no mercado profissionais aptos a trabalhar neste ambiente e nem mesmo programadores na academia. Para tanto, sugere-se o uso de um ambiente orientado à objetos.

Após a análise das linguagens orientadas à objetos, verifica-se que as duas possibilidades mais relevantes seriam o Java e o C# .NET. Dentre as duas opções o Java possui maior difusão na academia, porém a máquina virtual (VM) necessária para executar aplicações desenvolvidas nesta linguagem torna proibitivo o seu uso em computadores com menor poder computacional. Por outro lado, o C# .NET mesmo sendo uma linguagem que também gera programas com código intermediário (IL), necessitando o uso de VM para a execução dos programas, requer muito menos poder computacional.

Quando a usabilidade da linguagem em relação às outras, pode-se afirmar que o C# .NET é resultante do produto dos ambientes Java e C++. Portanto, como estes dois ambientes são bastante difundidos na academia e no mercado, quem têm bons conhecimentos em Java ou C++ tem grande possibilidade de adaptar-se à semântica da linguagem em questão, o C# .NET.

Então, conclui-se que hoje, em virtude do rápido avanço da tecnologia .NET devido à sua difusão no mercado e a facilidade do processo de desenvolvimento de aplicações através de sua utilização, sugere-se o seu uso para o desenvolvimento do aplicativo escopo deste trabalho de pesquisa.

Em se tratando de base de dados, é vital o uso de uma aplicação que pudesse efetuar a gerência das informações curriculares, que serão manuseadas no processo de apoio à gestão de competências. Neste contexto, segundo características observadas nos produtos Oracle, SQL Server e PostgreSQL, todos apresentaram características semelhantes, onde todos os três possuem suporte à XML, particularidade fundamental para este projeto de pesquisa, pois o sistema de Currículos Lattes trabalha com arquivos neste padrão. O fato de que os demais produtos não oferecerem suporte ao padrão XML, inviabiliza seu uso para o fim deste trabalho. Porém, dentre os três produtos um que se

destaca por ser software livre e ter ampla difusão nos ambientes acadêmicos e comerciais é PostgreSQL. Além disso, o fato de ser livre torna o banco de dados PostgreSQL relevante devido ao fato de que pode-se efetuar customizações no código-fonte do banco de dados, compilando um produto apenas com as características funcionais realmente necessárias ao projeto.

2.3 SISTEMAS INFORMATIZADOS PARA AUXÍLIO À GESTÃO DE COMPETÊNCIAS

O objetivo desta seção é o de analisar algumas das possibilidades em sistemas informatizados disponíveis no mercado e na academia, que se propões a efetuar a gestão de competências.

Neste contexto, serão apresentados o sistema e metodologias desenvolvidos e utilizados pela Siemens, a solução comercializada pela empresa Efix e utilizada por grandes empresas do mercado e o Sistema Lattes, composto pelo Currículo Lattes, Portal Inovação e sistema IS Curriculum. Esta seção resultará em um quadro onde estarão demonstrados elementos que auxiliarão na construção do protótipo que será discutida no Capítulo 3, onde será efetuada a simulação da ferramenta.

Conforme já discutido anteriormente nos itens 2.1.1 e 2.1.2, competências são propriedades emergentes dos indivíduos que resultam da articulação de conhecimentos, habilidades e atitudes, e mostram-se necessárias para por em prática ações produtivas.

Segundo Maclagan (1998) *apud* Flink (2003, p. 01), observando sob a ótica relacionada à postos de trabalho, o termo Gestão de Competências pode possuir diversos significados, sendo que em algumas definições se leva em conta o contexto do trabalho como as tarefas, os resultados e os erros e em outras, são descritas as características dos indivíduos que o realizam, como seus conhecimentos, habilidades e atitudes.

Para Borsatto (2006, p.03), a Gestão por Competências pode ser definida como um conjunto de práticas que têm por objetivo auxiliar ao gestor a diagnosticar os conhecimentos pessoais, experiências, o perfil psicológico (ou comportamental), a aptidão física e entre outros aspectos necessários a uma determinada função ou processo e um exercício crítico dessas competências no universo da instituição. Pereira (2002, p. 03) complementa que é necessário compreender que tipo de informações as organizações devem procurar manter, compreender e utilizar, para efetuar uma gestão efetiva de suas competências.

Neste sentido, segundo PEREIRA (2002, p.26):

[...] a Gestão de Competências auxilia a organização a conhecer melhor as potencialidades de cada pessoa que trabalha para a organização. Conhecer em detalhes o currículo dessas pessoas é importante, mas é uma informação que, mesmo sendo atualizada com frequência, revela muito pouco à organização sobre a capacidade de aprendizado, de mudança e de inovação dessas pessoas [...]

Já o autor Duizith *et. al.* (2003) trata de técnicas para a gestão de competências, e, afirma que estas têm grande possibilidade de auxiliar o gestor no processo de identificação das competências presentes em cada indivíduo, permitindo vislumbrar de uma melhor forma as orientações que o desenvolvimento humano deve tomar. O referido autor afirma que:

*As diversas metodologias e técnicas de Gestão de Competências permitem encontrar a pessoa certa para determinadas funções (dentro ou fora da organização) e ajudam a avaliar o indivíduo e o grupo, monitorando pontos fortes e fracos, para melhor aproveitamento e melhorar qualificação dos recursos. (DUIZITH *et. al.* 2003, p.02)*

Observando os sistemas que permitem efetuar gestão de pessoas e suas características, normalmente estes baseiam-se em avaliações onde as competências necessárias pela organização são apontadas e em seguida estas são buscadas nos indivíduos. Através desta análise, obtêm-se um mapeamento das competências necessárias e as disponíveis. (FERRAZ & LOPES, 2002) *apud* DUIZITH *et. al.* (2003, p.04) afirma que este processo facilita a alocação dos indivíduos em postos de trabalho, a formação adequada de equipes de trabalho e o planejamento para o desenvolvimento de recursos humanos.

Partindo do fato de que as organizações constantemente armazenam informações sobre pessoas, a vantagem da utilização de ferramentas informatizadas para a gestão deste tipo de informação é possibilitar ao gestor, uma melhor forma de minimizar o trabalho de análise e de classificação de competências técnicas e profissionais, buscando auxiliar no processo de gestão das competências necessárias às organizações.

Os sistemas informatizados de Gestão por Competências, são ferramentas que procuram auxiliar as organizações a gerenciar o “Capital Humano”¹⁸ das organizações, permitindo de forma automatizada de efetuar a gestão dos seus recursos humanos. Esta gestão pode englobar processos diversos principalmente ligados aos setores de recursos humanos como contratações, avaliações de desempenho, seleções para treinamentos, formação de equipes para projetos, busca por competências entre os colaboradores, dentre outros.

Neste cenário, algumas ferramentas informatizadas ganham destaque, como é o caso do conjunto de ferramentas disponibilizado pela empresa Efix. Estes sistemas vêm ganhando espaço no mercado corporativo e já foram adotadas por algumas das grandes empresas do cenário mundial, como por exemplo a BM&F, Grupo Votorantim, Global Village Telecom, Peugeot Citroën, Lear do Brasil, Hospital Israelita Albert Einstein, Editora Globo, 3M, ABN Amro Real, Coteminas, Cablettra do Brasil, Grupo VR, Gillete, NEC Brasil, Packard Bell, Polimold Industrial, Sabesp, SOS Computadores, Telemar, Unilever, Verisign, Inc, dentre outras 1200 empresas em cerca de 50 países, conforme pode ser observado em (Efix: Por que o Efix Avaliações, 2005, p. 02)

Faz parte do estudo o sistema Athena desenvolvido pela Siemens, com o objetivo de efetuar a gestão das competências de seus colaboradores. O grande diferencial da ferramenta da Siemens é ela foi modelada focando na gestão das competências de acordo com as necessidades transmitidas pelo planejamento estratégico organizacional, que acontece a cada cinco anos. Após o este planejamento, o sistema Athena que reflete a metodologia SCoM¹⁹ apresenta as necessidades de competências impostas pela demanda da organização.

O sistema Lattes é o sistema informatizado do CNPq composto por aplicativos, bancos de dados e sites na Web no qual são armazenados currículos das pessoas que formam a comunidade acadêmica como os pesquisadores, estudantes, professores, dentre outros. Este sistema foi concebido pelo CNPq, através de um levantamento com seus consultores visando além da facilidade de manutenção dos dados, criar um padrão que

¹⁸Segundo Seiffert (2002, p.53) Capital Humano é a capacidade, conhecimento, habilidade, experiência individual, criatividade e capacidade de inovar dos colaboradores de uma organização.

¹⁹ A metodologia SCom/Siemens é discutida em 2.1.4.2.

atendesse as necessidades de fomento como planejamento em C&T (PORTAL LATTES, Diretório de Grupos de Pesquisa, 2007). O primeiro protótipo foi lançado em março de 1999.

Com o passar do tempo, mais instituições foram aderindo à padronização do sistema em virtude das funcionalidades oferecidas. Culminou que a partir de 2002, todos os bolsistas de pesquisa, mestrado, iniciação científica, professores e orientadores, precisam ter cadastrado o seu currículo Lattes no CNPq, sendo que a inexistência do mesmo resulta em cancelamento de pagamento de bolsas e projetos; também é obrigatório para os participantes dos diretórios de grupos de pesquisa. Com isso, a base Lattes tornou-se um grande repositório de dados, contendo informações relevantes como linhas de pesquisa, trabalhos em andamento e grupos de pesquisa em produção do país.

Na seqüência, os itens 2.3.1, 2.3.2 e 2.3.3 irão apresentar os três sistemas informatizados

2.3.1 Sistema Efix Competências

O pacote de aplicativos desenvolvido pela empresa Efix é compatível com as plataforma Internet ou Intranet (roda através de navegadores como o Internet Explorer), estando dividida em dois conjuntos de ferramentas distintos, nominados “*Efix Performance Suite*” e “*Efix Survey Suite*”.

O primeiro, o *Efix Performance Suite* é conjunto de sistemas informatizados que têm por objetivo estabelecer processos de verificação de desempenho individual e gerencial, buscando alinhar as atividades diárias de funcionários e equipes com as metas e objetivos corporativos. Suas ferramentas, oferecem soluções que possibilitam efetuar avaliações sob as perspectivas quantitativas e qualitativas de cada funcionário buscando maximizar seu desempenho, além de orientar os esforços em treinamentos, buscando otimizar o desenvolvimento dos colaboradores e das equipes. O objetivo da segunda ferramenta, o *Efix Survey Suite*²⁰ é o de medir indicadores de clima organizacional que estariam vinculados ao objetivo do negócio da empresa, onde são quantificados os aspectos ligados

²⁰ Como o processo de análise dos fatores tratados pela ferramenta *Efix Survey Suite* não faz parte do escopo deste trabalho, o foco deste capítulo é mantido sobre o módulo da ferramenta *Efix Performance Suite* que permite a análise de competências, doravante intitulado *Efix Competências*.

a liderança, processos e gestão. Quanto a liderança, são analisados fatores tais como reconhecimento, apoio, feedback, planos de carreira e processo de decisão. Quanto aos processos são analisadas principalmente políticas e normas de recursos humanos aplicadas pela empresa. Na gestão são verificados indicadores quanto à missão, visão, valores, futuro, imagem, e inovação na organização.

O *Efix Performance Suite* divide-se em um conjunto de quatro sistemas distintos, *Efix Competências*, *Efix Desempenho*, *Efix Mapeamento* e *Efix Certificação*. Como o escopo deste capítulo teórico é demonstrar ferramentas existentes que permitam a gestão de competências, então é abordado apenas o mecanismo de análise da ferramenta *Efix Competências*.

Através da análise de competências por meio da ferramenta *Efix Competências*, o gestor define as competências existentes e opcionais para avaliação em seus colaboradores. Busca-se encontrar no colaborador as competências existentes por meio de questionários pré-definidos que deverão ser respondidos conforme a metodologia em 360 graus apresentada no item 2.1.4.3 deste trabalho, ilustrado pelas figuras demonstradas na seqüência²¹. (Efix: Monte seu projeto de avaliação - Competências, 2005 p. 8-10).

A Figura 17 apresenta uma tela onde são valorados de 0 a 5 características de comportamentos e habilidades do colaborador em questão, de atuar em situações corriqueiras que podem ocorrer em curto, médio e longo prazo.

²¹ Nestas figuras são demonstradas algumas das telas componentes do sistema Efix Competências, que estão no manual eletrônico do sistema, que está disponível no site referenciado no texto, na língua inglesa. Como não é objetivo deste trabalho verificar toda a semântica das palavras quando de sua tradução, optou-se por apresentar as telas em seu idioma original.

QUESTIONS						
QUESTIONS	0	1	2	3	4	5
1 Addresses current opportunities and problems by improving and adapting existing approaches.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2 Identifies and exploits opportunities in the short-term. Applies own knowledge and expertise to developing new approaches to exploiting these.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3 Able to spot opportunities and problems in the medium-term (3 to 12 months), and develops new approaches, products and services based on these. Understands the business case for diversity and equal opportunities. Uses networks to seek fresh ideas. Encourages different agencies to pool resources and expertise.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4 Sees and acts on long-term (beyond 12 months) opportunities and problems. Formulates completely new and 'off the wall' ideas and concepts, which create a potential opportunity, and uses innovative approaches to allow them to be realised. These may address underlying needs that have not yet been identified by customers.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	0	1	2	3	4	5

Figura 17 – Questionário de Identificação de Competências. Fonte: Portal Efix

A figura na seqüência demonstra a forma através da qual o sistema procura explorar o contexto empreendedor do indivíduo, buscando verificar sua possibilidade de responder frente às dificuldades, atingir metas, etc. Nota-se que a resposta neste momento não corresponde a um valor, devido ao fato de que o comportamento é um elemento tácito e de delicada mensuração.

ACHIEVEMENT	
Achievement is about having the sustained energy and determination in the face of obstacles to set and meet challenging targets, in compliance with quality, time and diversity standards, and delivering the required business results.	
QUESTIONS	
1	Achievement
<input type="radio"/>	WANTS TO DO A GOOD JOB AND DOES NOT GIVE UP Clarifies what is expected. Is positive and enthusiastic about the job. Does not give up at the first obstacle. Respects the rights of others whilst achieving one's own goals.
<input type="radio"/>	WORKS TO GOALS AND MANAGES OBSTACLES Understands and works towards goals set by others. Measures progress against targets. Seeks to understand reasons for obstacles and to find ways to overcome. Acknowledges the work and contribution of others.
<input type="radio"/>	DRIVES FOR IMPROVEMENT DESPITE FREQUENT OBSTACLES Is determined despite frequent obstacles. Tackles difficult problems and takes personal responsibility for reaching solutions. Seeks ways to improve overall performance levels to give higher levels of satisfaction to target groups.
<input type="radio"/>	SETS AND MEETS CHALLENGING GOALS & SEEKS LONG TERM Achieves significant progress in the long term, wider performance of the Council. Sets out to be the best – has own measures of excellence and works to these.
<input type="radio"/>	ASSESSES AND TAKES RISKS TO MAKE SIGNIFICANT GAINS Makes decisions through weighing up the cost-benefit and risk implications. Acts entrepreneurially to make performance gains.

Figura 18 – Avaliação por Competências – Perfil do Colaborador. Fonte: Portal Efix

Neste momento, é montada a matriz das competências coletadas pelo sistema, onde são informadas as competências obrigatórias e sua ordem de relevância perante as demais. Estas informações são necessárias para poder efetuar a segregação dos dados coletados pelos colaboradores para o posto de trabalho, que neste caso é o de gerente.

Observam-se através da Figura 19, que são buscadas competências identificadas e quantificadas nesta ordem de relevância: item 1: conclusividade²² entre 2 e 3 indícios; item 2: raciocínio analítico entre 3 e 4 indícios; item 3: orientação de serviços à clientes entre 3 e 4 indícios; item 4: trabalhar estrategicamente entre 3 e 4 indícios; item 5: empreendedorismo entre 2.2 e 3 indícios.

²² O termo original *achievement* no idioma inglês significa façanha, realização de empreendimento. Porém o entendimento dado ao termo é o de possuir a capacidade de finalização de projetos, concluir atividades iniciadas.

COMPETENCIES EXPECTED FOR THE JOB POSITION MANAGER			
ORDER	MANDATORY COMPETENCIES		COMPETENCY
	MIN. EXPECTED	MAX. EXPECTED	
1	2	3	Achievement
2	3	4	Analytical Thinking
3	3	4	Customer Service Orientation
4	3	4	Working Strategically
5	2.2	3	Entrepreneurship
6	0	0	Entrepreneurship

UPDATE

Figura 19 – Quantificação da Competência - Índicios. Fonte: Portal Efix

Ao final da definição das competências necessárias ao posto de gerente coletadas através do referido sistema, conforme demonstra a Figura 20 é apresentado o resultado total do processo onde são demonstradas as competências pesquisadas na ordem de relevância definida anteriormente. Na coluna de resultado, os valores de escala definidos na Figura 19 são demonstrados através da linha amarela, sendo que a linha vermelha indica valor abaixo da expectativa e a verde acima da expectativa.

Observa-se que conforme demonstrado na Figura 20 que, todos os candidatos pesquisados possuem as três primeiras competências que são empreendedorismo, raciocínio analítico e orientação de serviços a clientes. Porém do conjunto analisado nenhum conseguiu obter o índice necessário às duas últimas competências ranqueadas, que são trabalhar estrategicamente e empreendedorismo corporativo, que é a capacidade de enxergar novas oportunidades e negócios.

Este cenário demonstra que o gestor precisará atuar no processo para verificar quais os colaboradores que estiveram mais próximos do índice apresentado pelo sistema como ideal e efetuar uma nova avaliação destes candidatos.

Outra possibilidade seria reajustar os índices definidos, conforme apresentado na Figura 19 e gerar novamente o relatório de resultados, para localizar os indivíduos que mais se aproximaram do ponto de ideal das competências necessárias ao posto de gerente.

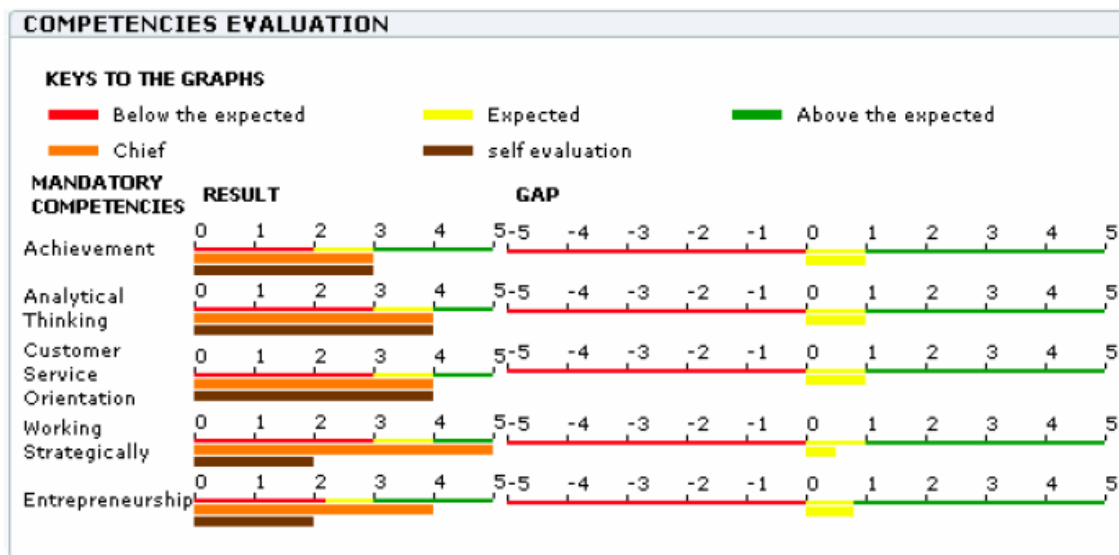


Figura 20 – Avaliação de Competências - Resultado. Fonte: Portal Efix

O segundo pacote de sistemas citado, o “Efix Survey Suite” são os sistemas que permitem aos gestores buscar e apresentar indicadores que demonstram o “Clima Organizacional”, conceito que a própria empresa fabricante do sistema define como:

“[...] a percepção coletiva que os funcionários têm da corporação [...]”

e afirma que:

“[...] e sua a melhoria contribui para a melhoria de fatores críticos, como motivação, turnover, performance organizacional e processos de mudança [...]”

2.3.2 Sistema Athena

O sistema Athena é uma ferramenta que foi concebida inicialmente para atender as demandas de mapeamento de competências das áreas de pesquisa, desenvolvimento e engenharia da Siemens do Brasil. Posteriormente, foram agregadas à ferramenta a capacidade de permitir ao gestor identificar quais competências estão vinculadas a cada colaborador da empresa e estabelecendo métricas que também podem apresentar a forma através da qual estas competências estão distribuídas na da organização como um todo conforme demonstrado pela Figura 21.

The image shows two overlapping browser windows from Microsoft Internet Explorer. The left window displays the 'Sistema Athena' search interface. The right window displays the user profile information for 'BARROS WILLIAM'.

Informação sobre o usuário

Dados do Usuário

- Nome: BARROS WILLIAM ([link para o SCD](#))
- Setor: ICN C WN DP
- Descrição da área de trabalho: Gerente do grupo de Project Management do WN R&D (Desenvolvimento/Comutação Pública). Planejamento e acompanhamento de projetos (custos, prazos, riscos, requisitos, métricas de qualidade, etc).
- Idiomas que fala: Alemão(***), Inglês(***), Português(****);

Habilidades do Usuário

- ◆ Conhecimento Geral Gerenciamento Riscos(**)
- ◆ Desenv. Software Parametrização Linhas(**)
- ◆ Desenvolvimento de Software LIP(**)
- ◆ Desenvolvimento de Software Trunkscriber(**)
- ◆ Diretriz SAT(***)
- ◆ Ferramenta INP(***)
- ◆ Ferramenta MS Project(**)
- ◆ Ferramenta SAT(***)
- ◆ Gerenciamento Projetos Lei 10.176/01(*)
- ◆ Gerenciar Desenvolvimento em Paralelo(***)
- ◆ Linguagem Assembler (***)
- ◆ Linguagem C(**)
- ◆ Linguagem JAVA(**)
- ◆ Linguagem PASCAL(***)
- ◆ Modelagem de Processo(*)
- ◆ Planejamento Projeto de Desenvolvimento(***)
- ◆ Processo INP(***)
- ◆ Processo INT(*)
- ◆ Processo PPP:C(***)
- ◆ Projetos de Modalqem de Processo(*)

Buscar Conhecimento

Escolha um item ao lado: habilidades

Entre com a palavra-chave para busca:

lei de TI

Resultados da Busca

Mostrando Resultados de 1 a 3 de 3:

- B&A Siemens Ltda. (Brasil)
 - Gerenciamento de Negócios
 - Gerenciamento dos Fundos do Negócio
 - Gerenciar os Fundos Incentivados
 - Sistema SGPI
- P&D Siemens Ltda. (Brasil)
 - Gerenciamento de Projeto
 - Gerenciamento Integrado do Projeto
 - Gerenciar através de Planos Integrados

Figura 21 – Competências no Sistema Athena

Desenvolvida pela Siemens Corporate Technologies juntamente com a Pontifícia Universidade Católica do Paraná e o Centro de Estudos Avançados do Recife, o sistema Athena foi rapidamente estendido às outras unidades de negócio da Siemens, inclusive à unidades Siemens de outros países. A Figura 22 mostra a tela de entrada do Sistema Athena.

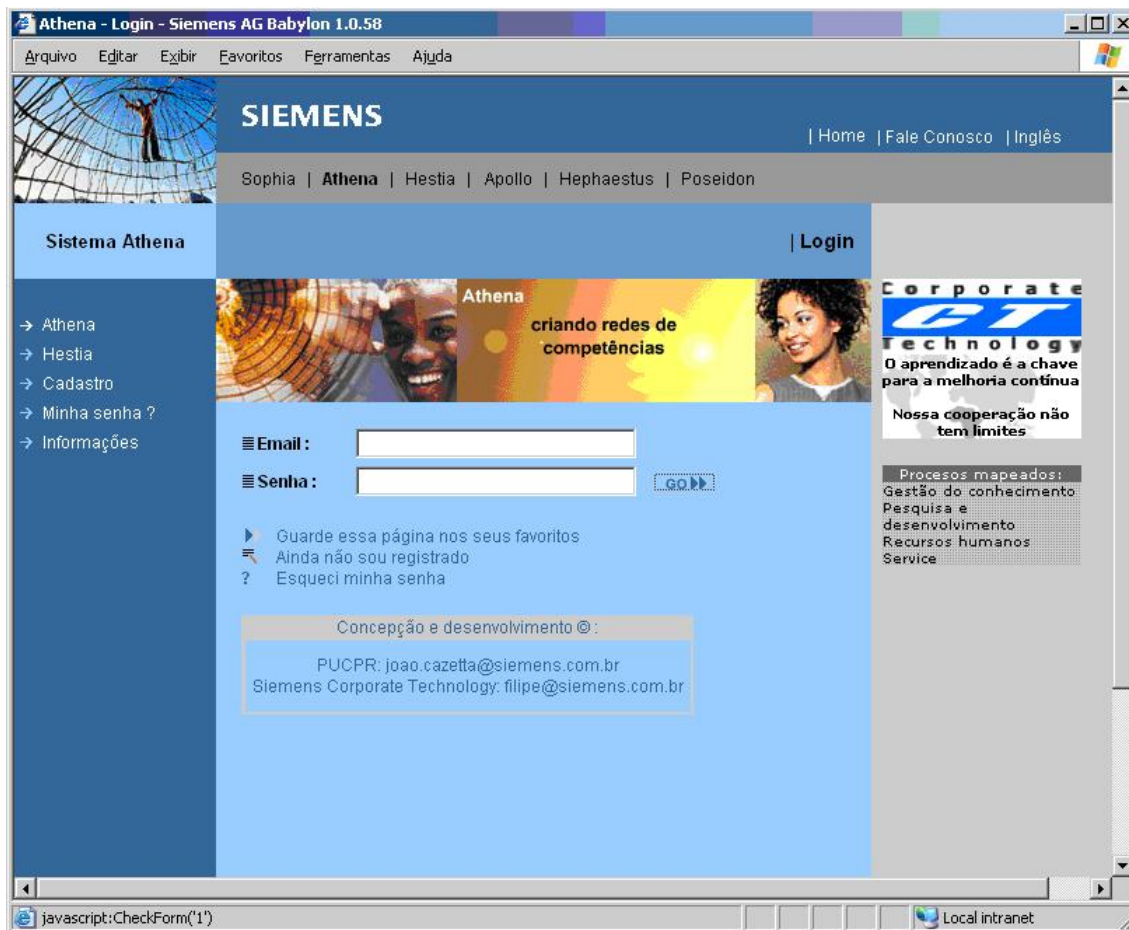


Figura 22 – Sistema Athena

A ferramenta Athena utiliza a metodologia SCoM²³, como pode ser observado através da Figura 23, onde na inserção das habilidades do colaborador, o sistema informatizado aponta para as habilidades SCoM disponíveis.

²³ A metodologia SCoM é tratada no item 2.1.4 deste trabalho, à página 19.

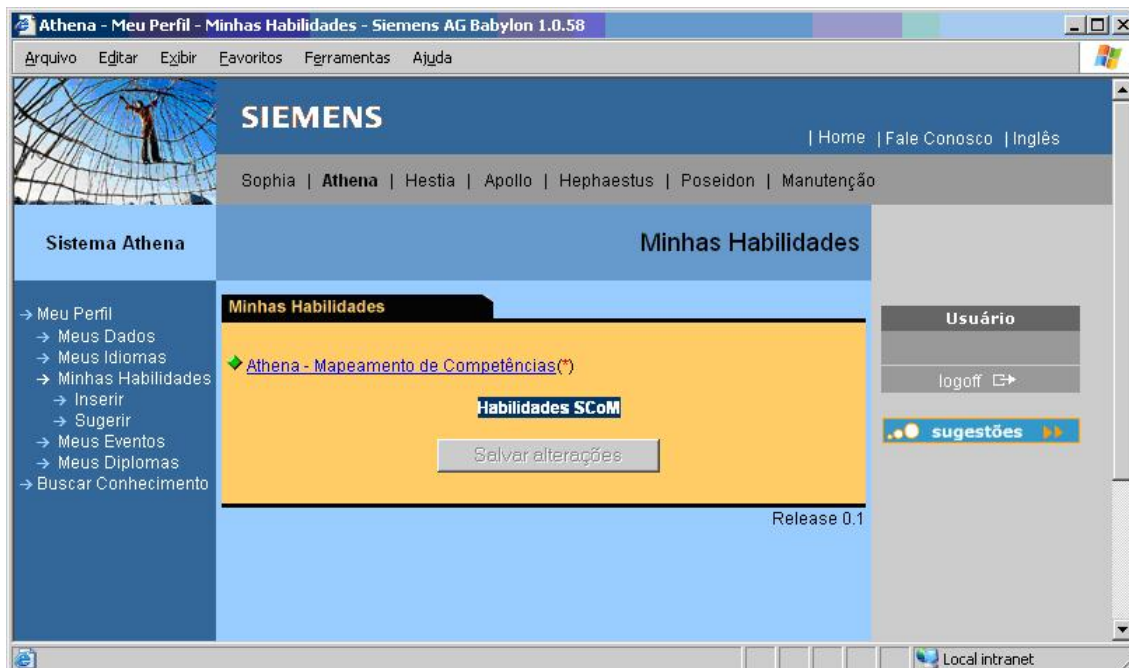


Figura 23 – Habilidades SCoM

Estas habilidades são atribuídas ao colaborador durante o cadastro de seus dados pessoais. Juntamente neste processo de cadastramento, o colaborador também informará ao sistema Athena as suas habilidades, que seguem as diretrizes propostas pela metodologia SCoM, e apresentadas pela Figura 24.

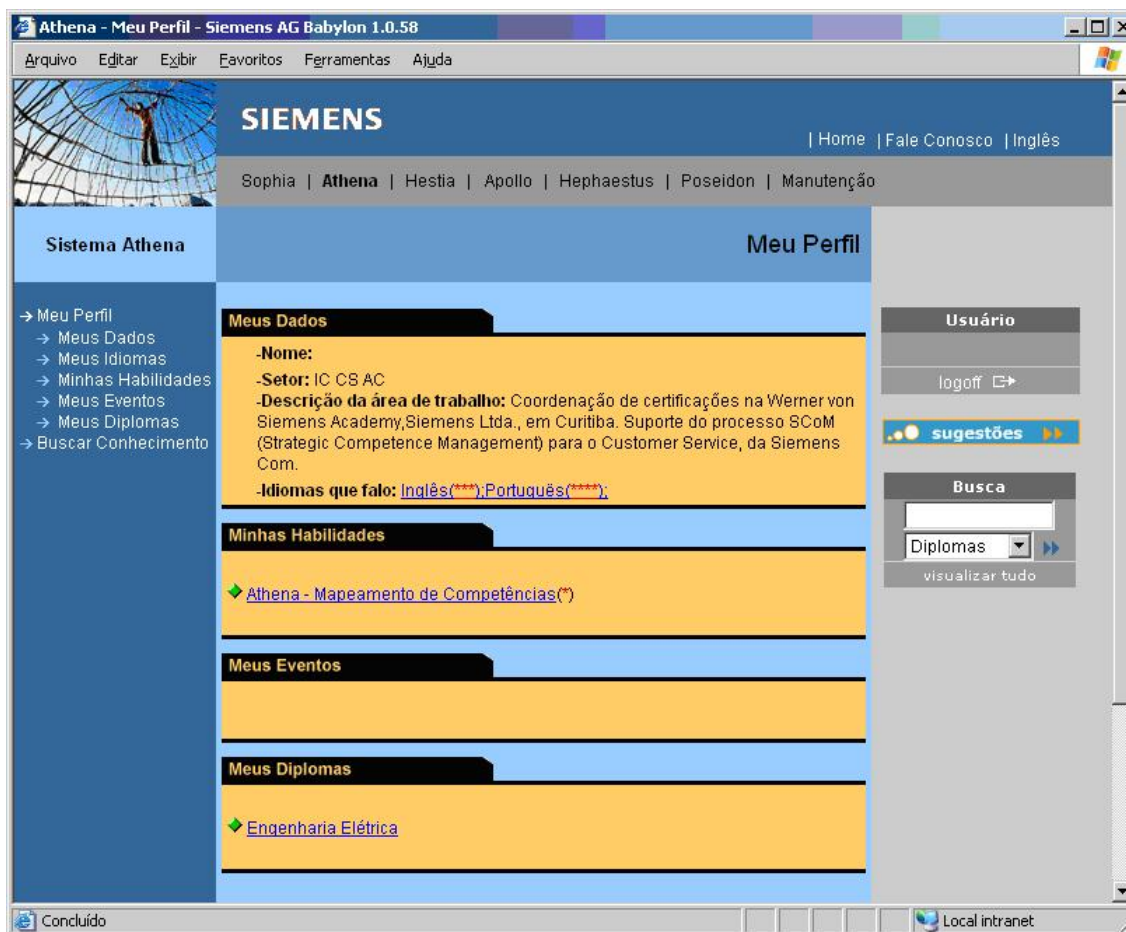


Figura 24 – Cadastro de dados no Athena

Mais tarde, é através destas informações que é efetuada a gestão das competências da Siemens, permitindo aos colaboradores indiferentemente da unidade Siemens que estejam vinculados, tenham a possibilidade de compartilhar suas competências dentre todas as unidades de negócio da Siemens Mundial.

A duas versões do sistema Athena foram desenvolvidas utilizando ferramentas Microsoft. Na primeira versão, que não possuía integração com os demais sistemas da empresa, foi utilizada a ferramenta Microsoft Access para gerenciar os dados e a implementação da interface foi na linguagem ASP. Já na segunda versão, o sistema é integrado aos sistemas de gestão, os dados são gerenciados pelo sistema gerenciador de banco de dados Microsoft SQL Server e a linguagem de implementação da interface é o ASP.NET.

2.3.3 Sistemas Informatizados do Instituto Stela

No contexto deste trabalho, foi dada uma maior ênfase no estudo do sistema de informações curriculares Currículo Lattes, devido ao fato de que este sistema é utilizado posteriormente para a coleta das informações que serão utilizadas pelo sistema agentes no Capítulo 3 deste trabalho.

O Instituto Stela é uma entidade sem fins lucrativos criada em 2002, que atua em pesquisa e desenvolvimento nas áreas de Tecnologia da Informação e Engenharia do Conhecimento. Este instituto tem como raízes o Grupo Stela, criado no Programa de Pós Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina no ano de 1995 (REVISTA INFORME, p. 06).

Dentre os projetos desenvolvidos pelo Instituto Stella, pode-se citar a Plataforma Curricular Lattes, o sistema IS Curriculum e o Portal Inovação.

A Plataforma Curricular Lattes tem por objetivo a coleta e organização de informações curriculares, sendo que é utilizado em sua grande maioria por professores e pesquisadores que se encontram na academia. Os dados obtidos através desta plataforma são utilizados pelos demais sistemas desenvolvidos pelo instituto. Segundo o próprio Instituto Stela através de censo realizado em 2005, este sistema possuía aproximadamente 605 mil currículos, 20 mil grupos de pesquisa e 335 instituições de ensino cadastrados, o que gerou mais de 13 milhões de acessos em 5 anos de operação.

O sistema IS Curriculum é um sistema informatizado que permite a busca e a análise de informações curriculares de pessoas que estão vinculados a uma organização, demonstrando suas linhas de pesquisa, principais palavras-chave de seus trabalhos e competências potenciais.

Já o Portal Inovação é um sistema baseado em portais que procura aproximar as empresas da universidade, possibilitando que as empresas busquem através de palavras-chave, por competências de pesquisadores na base curricular da Plataforma Lattes.

Plataforma Curricular Lattes

A Plataforma Curricular Lattes é a base de todos os demais sistemas desenvolvidos pelo Instituto Stela apresentados neste trabalho, conforme demonstra a Figura 25. Ele é um conjunto de sistemas de informação, composto por bases de dados e portais Internet voltado para a gestão de informações curriculares. Esta plataforma foi

desenvolvida para o CNPq com o intuito de racionalizar o processo de gestão de informações curriculares e possibilitar a integração dos sistemas de informação das agências federais de fomento e demais instituições de ensino e pesquisa.

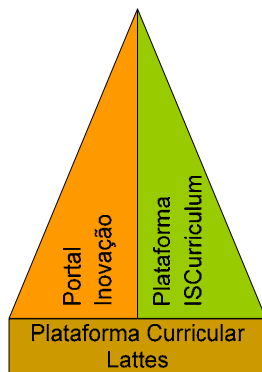


Figura 25 – Composição dos Sistemas Lattes

Possui dentre seus objetivos principais, permitir o intercâmbio de informações curriculares na comunidade científica. Pode-se, por exemplo, localizar um pesquisador e saber em qual área de concentração este está trabalhando. Com este potencial de informação, há a possibilidade de se obter dados estatísticos bastante interessantes, como por exemplo, para onde caminha a produção técnico/científica nacional, quais as linhas de pesquisa mais relevantes de cada instituição de ensino, quais são os grupos formais de pesquisadores em produção, etc.

Os dados que compõe a plataforma Lattes são obtidos através do preenchimento de um questionário eletrônico diretamente no site do CNPq através do endereço <http://lattes.cnpq.br>, conforme demonstra a Figura 26, a tela de cadastro inicial do Currículo Lattes.

The image shows a web browser window titled "Cadastro de novo RH - Mozilla Firefox". The address bar displays the URL: https://www.cnpq.br/sigef_imp/owa/PKG_CAD_RH.GN_TERMO.... The page header features the CNPq logo and the text "Conselho Nacional de Desenvolvimento Científico e Tecnológico".

The main content area is a registration form with the following sections:

- Identificação**
 - Nome completo *
 - Nacionalidade: Brasileira Estrangeira
 - CPF *
- Dados do nascimento**
 - Pais de Nascimento: [dropdown menu]
 - Data *
- Endereço Eletrônico**
 - E-mail
 - Senha
 - Confirme a Senha

At the bottom of the form, there are two buttons: "Confirmar" and "Não Confirmar".

The footer of the page contains the text: "SOBRE O CNPq | FOMENTO | PLATAFORMA LATTES | ATENDIMENTO CNPq | SERVIÇOS CNPq | OPORTUNIDADES" and "MCT | GOVERNO FEDERAL". The status bar at the very bottom shows "Concluído" and the website address "www.cnpq.br".

Figura 26 – Currículo Lattes - Cadastro inicial

Após o aceite do termo de adesão para a utilização do sistema, é realizado o cadastro dos dados básicos e da senha para o acesso ao sistema. Em seguida, o site do CNPq redireciona o indivíduo para a tela de entrada do portal demonstrado pela Figura 27, onde o restante das informações curriculares deverão ser cadastradas. Após efetuado o cadastro, é apresentado um *resumé* do currículo na tela inicial, juntamente com as ferramentas que serão utilizadas para interagir com o sistema do portal.



Figura 27 – Tela inicial do Currículo Lattes

Dentre os dados coletados pelo sistema, estão além dos dados pessoais as informações sobre formação acadêmica e titulação, produções bibliográfica e técnica, orientações e supervisões, produção cultural, participação em eventos e bancas. A Figura 28 demonstra as informações quanto a formação acadêmica e titulação cadastradas em um Currículo Lattes.

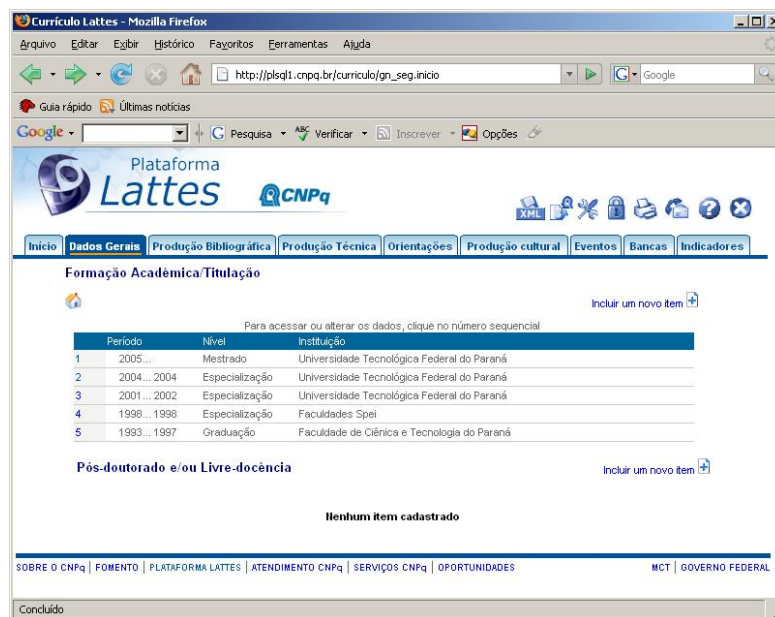
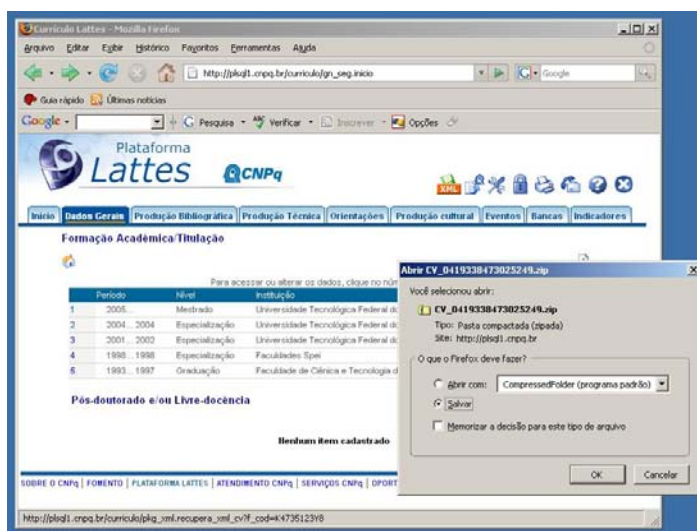


Figura 28 – Formação Acadêmica e Titulação

Um recurso extremamente relevante do sistema de Currículos Lattes é a possibilidade de exportar as informações curriculares diretamente para o formato XML²⁴, como pode ser observado na Figura 29. O botão que inicia o processo de exportação aparece em laranja no lado superior direito da figura.



²⁴ A estrutura de dados XML é discutida no item 2.2.2 deste trabalho.

Figura 29 – Exportação de currículo em formato XML

Este mecanismo permite que as informações curriculares geradas pelo sistema de Currículos Lattes possam ser lidas por outros sistemas que possuam compatibilidade com o padrão XML.

Plataforma IS Curriculum

A plataforma de software IS Curriculum desenvolvida pelo Instituto Stela é um conjunto de softwares que permite efetuar consultas nas informações cadastradas na base curricular Lattes Institucional. Para habilitar a sua utilização, a instituição deve instalar e manter um banco de dados semelhante ao utilizado pelo Sistema Lattes no CNPq, que deverá ser utilizado pelo portal institucional que faz parte do sistema.

Para alimentar com currículos esta base institucional, os pesquisadores vinculados à instituição precisam instalar em seus computadores um software fornecido pelo Instituto Stela chamado de *PLI-CV*. Este software permite o envio do currículo Lattes simultaneamente para o CNPq e para o banco de dados institucional que proverá funcionalidade à plataforma IS Curriculum, conforme demonstra a Figura 30 e Figura 31.

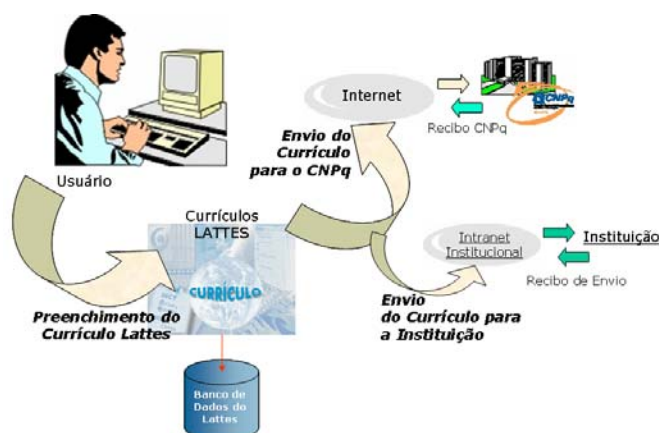


Figura 30 – Envio de Currículo para o IS Curriculum (Fonte: MARCHEZAN, 2006)

Desta forma, o processo de envio do currículo para ambas as bases de dados, a do CNPq e a institucional tornam-se transparentes para os usuários do sistema. Quando o currículo é enviado pela Intranet institucional, é através de um sistema chamado “Guichê de Recepção” que o currículo é recebido no sistema Lattes Institucional. Após a recepção deste currículo, é gerado um protocolo para o usuário e o currículo recebido no formato XML é

encaminhado para o “Sistema de Carga” que irá gravá-lo na base de Currículos Lattes Institucional. O acompanhamento de todo o processo de recepção e carga de dados é efetuado por um módulo do sistema chamado de “Sistema de Acompanhamento”.

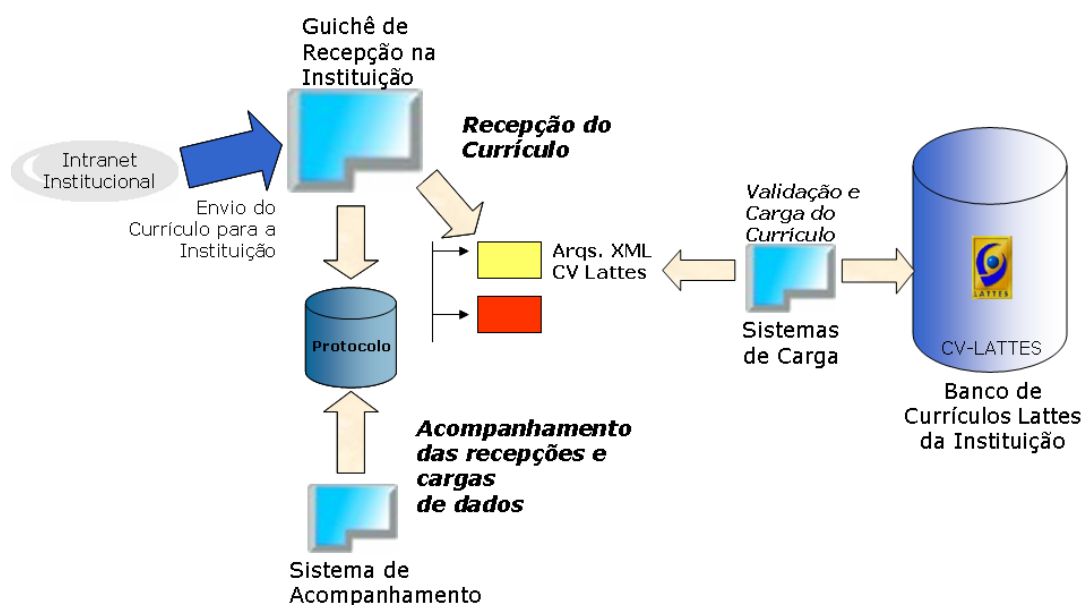


Figura 31 – Recepção de Currículo Lattes no IS Curriculum (Fonte: MARCHEZAN, 2006)

Uma vez gravado na base de Currículos Lattes Institucional, o currículo do usuário estará disponível para ser consultado na instituição pelo portal IS Curriculum. Através deste portal, podem ser executadas diversas operações de visualização, classificação, censos, acompanhamentos, dentre outras operações conforme demonstra a Figura 32.

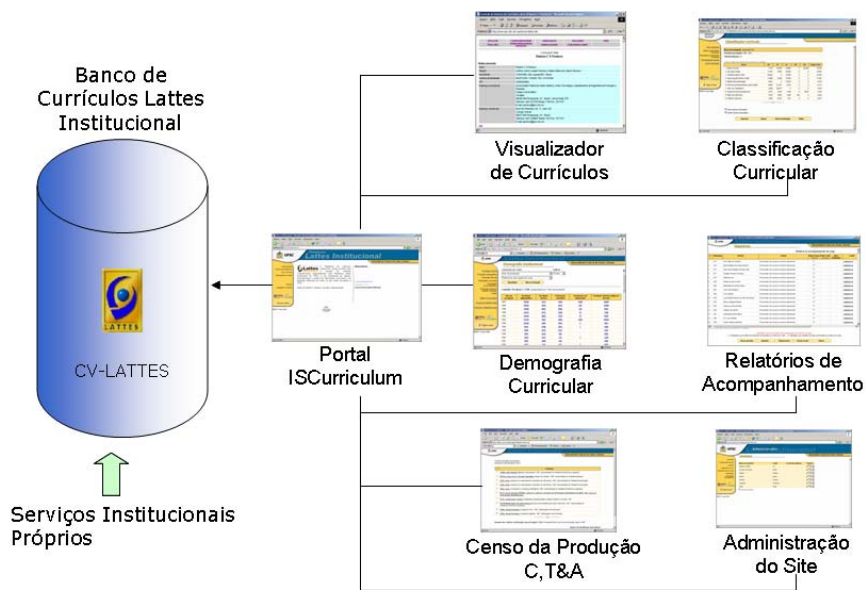


Figura 32 – Serviços do Portal IS Curriculum (Fonte: MARCHEZAN, 2006)

Neste contexto, a área do portal IS Curriculum que permite a identificação de profissionais de acordo com um perfil curricular previamente definido é o “Visualizador de Currículos”, apresentado pela Figura 33. Através dele podem-se efetuar buscas nas bases curriculares para localizar profissionais que poderiam lecionar em determinadas áreas do conhecimento, quais atividades estes docentes desenvolvem, quais são suas produções bibliográficas, onde estudaram e demais informações relevantes.

The figure displays three overlapping screenshots of the 'Lattes Institucional' web application, which is used for managing and visualizing curriculum data at the Universidade Federal do Rio Grande do Sul (UFRGS).

The top screenshot shows the main header of the application, including the UFRGS logo and the title 'Plataforma Lattes Institucional'. It also features a navigation menu and a 'Últimas Notícias' section.

The middle screenshot shows the 'Ver currículos' page, which allows users to search and filter curriculum data. It includes a sidebar with navigation options and a main content area with search filters and a list of users.

The bottom screenshot shows a detailed profile for Angela Freitag Brodbeck, including personal data, academic titles, and a list of professional titles.

Nome	Título	Atuação profissional
Angela Freitag Brodbeck	Doutorado	angela@cc.colibec.com.br
Antonio Domingos Padula	Doutorado	adpadula@ufrgs.br
Carlos Alberto Martins Calvoeira	Doutorado	es15881@ufrgs.net
Carlos Alberto Vargas Rossi	Doutorado	carlosar@ufrgs.br
Carriem Luiza Yoshida Orsini	Doutorado	carriem@ufrgs.br
Claudio Pinho Mazzilli	Doutorado	cmazzilli@ufrgs.br
Cristiane Pinheiro dos Santos	Doutorado	cpinheiro@ufrgs.br
Denis Dorcoteles	Doutorado	denis@ufrgs.br
Eli Moisés Francisco	Doutorado	emfrancisco@ufrgs.br
Eduardo Elias Saito	Doutorado	esaito@ufrgs.br

Resumo de seleção de critérios: Seleção de currículos cujas Instituição de atuação, orientação ou formação seja UFRGS - Universidade Federal do Rio Grande do Sul, na área Administração, com orientação em andamento em mestrado ou doutorado.

Imprimir apenas a página atual
 Salvar apenas a página atual

[Nova busca](#)
[Imprimir](#)
[Salvar](#)
[Configurar](#)

Figura 33 – Visualizador de Currículos (Fonte: MARCHEZAN, 2006)

O módulo do sistema IS Curriculum que permite efetuar simulações para demonstrar qual é a atual posição em indicadores no que se refere a projetos, produção científica e tecnológica e outras informações que compõe o perfil institucional é a Demografia Curricular. Através dela pode-se levantar indicadores quantitativos que demonstram, por exemplo, qual é a produção em determinada área do conhecimento, etc.

The image shows a web browser window displaying the 'Demografia Institucional' page of the Universidade Federal do Rio Grande do Sul (UFRGS). The page is titled 'Demografia Institucional' and shows the 'Unidade de análise: Produção C,T&A - Produção bibliográfica'. There are search filters for 'Área de produção' and 'Filtros'. A table shows the number of articles published in various areas, with 'Biologia Geral' highlighted. A detailed view of a production is shown, including the author 'OLIVEIRA, Luis Paulo Luna de; BODMANN, Bardo Ernst Josef; CECHIN, Adelmano Luis' and the title 'Um Modelo Fractal para a Geração de Sons Pulmonares de Alta Frequência'. Below this, a list of productions is shown, with the first one being 'Um Modelo Fractal para a Geração de Sons Pulmonares de Alta Frequência' by Adelmano Luis Cechin in 2001.

Área de produção	Artigos publicados no país	Artigos publicados no exterior	Trabalhos publicados	Livros publicados
Wção Informada	9066	2884	14576	187
Administração	1282	76	2274	15
Administração Hospitalar	2	3	6	-
Administração Rural	1	-	-	-
Agronomia	6299	684	5122	39
Antropologia	632	84	228	7
Arqueologia	276	28	165	31
Arquitetura e Urbanismo	438	129	894	11
Artes	659	97	452	13
Astronomia	54	692	223	16
Biofísica	72	169	48	4
Biologia Geral	184	159	85	24

#	Informado por	Título	Ano	País
1	Adelmano Luis Cechin	Um Modelo Fractal para a Geração de Sons Pulmonares de Alta Frequência	2001	Brasil
2	Andre Antonio Souto	Trichomonas vaginalis: Microtubule Cytoskeleton Distribution Using Fluorescent Taxoid	2002	Brasil
3	Arno Rudi Schwantes	Amido Hidrolisado de Milho Como Suporte Eletroretico	1981	Brasil
4	Arno Rudi Schwantes	An Improved Technique For Rapid Identification Of Hemoglobin Chains By Starch Gel Electrophoresis	1970	Brasil
5	Carolina Casagrande Blanco	Funções de Autocorrelação em Sistemas Fluídos Longe do Equilíbrio: Um Protótipo do Fenômeno de Autocorrelação em Sistemas Naturais	2001	Brasil
6	Clarice Luz	Atividade Natural Killer: Implicações na Saúde e nas Doenças Humanas.	1995	Brasil
7	Clarice Luz	Relação entre a Evolução Clínica na Depressão Maior e Atividade do Sistema Imune.	1995	Brasil
8	Cristina Pungastri	Further phenotypic characterization of the pso mutants of Saccharomyces cerevisiae with respect to DNA repair and response to oxidative stress.	2001	Brasil

Figura 34 – Demografia Curricular (Fonte: MARCHEZAN, 2006)

O sistema que permite efetuar a classificação quantitativa das pessoas ligadas à instituição, de acordo com sua produção C,T&A é o Sistema Classificador Curricular demonstrado pela Figura 35. Através dele podem ser respondidas questões como quem são os pesquisadores com maior produção, quem são os alunos com mais experiência científica, etc.

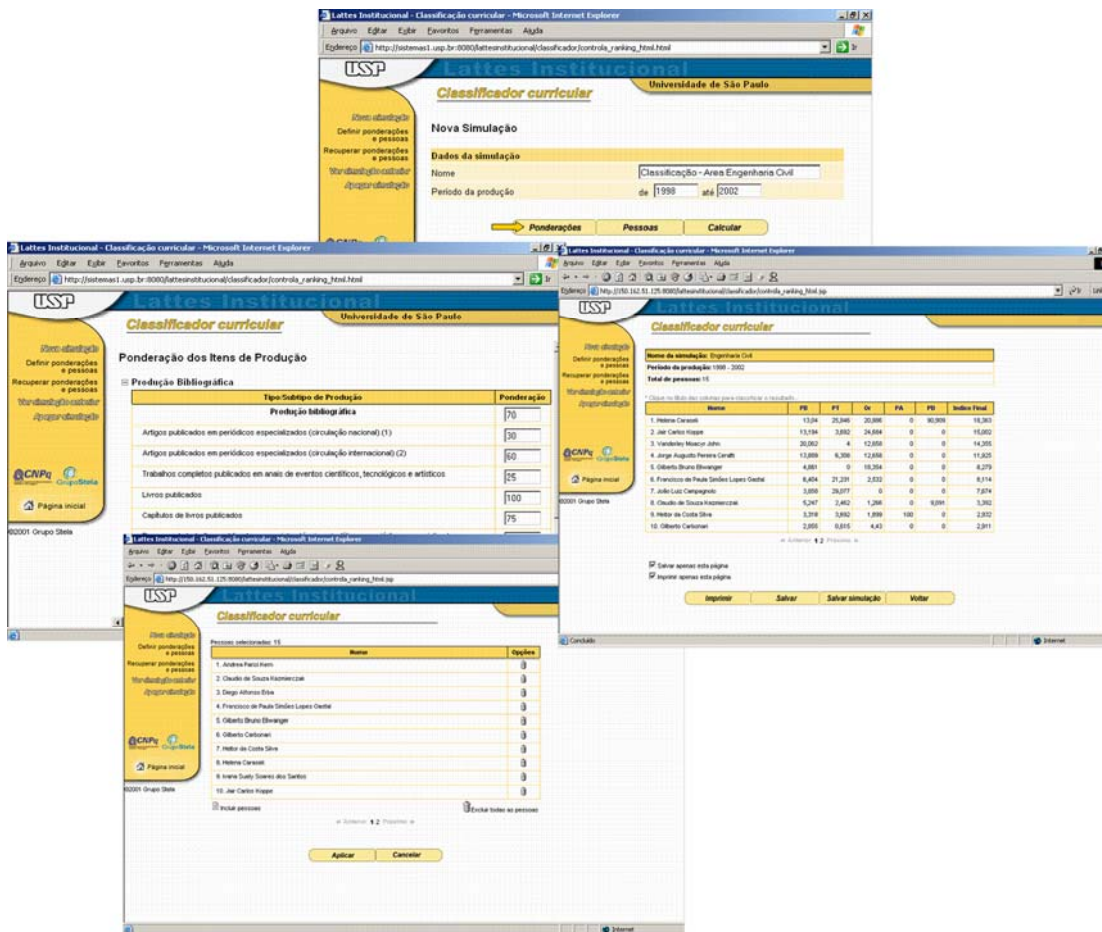


Figura 35 – Classificador Curricular (Fonte: MARCHEZAN, 2006)

Portal Inovação

O Portal Inovação é uma ferramenta que também foi desenvolvida pelo Instituto Stela e disponibilizada para a comunidade pelo CNPq, conforme demonstra a Figura 36. Nele, as empresas, profissionais especialistas ou ICTIs – Instituições de Ciência, Tecnologia e Inovação podem registrar-se para participar das buscas.

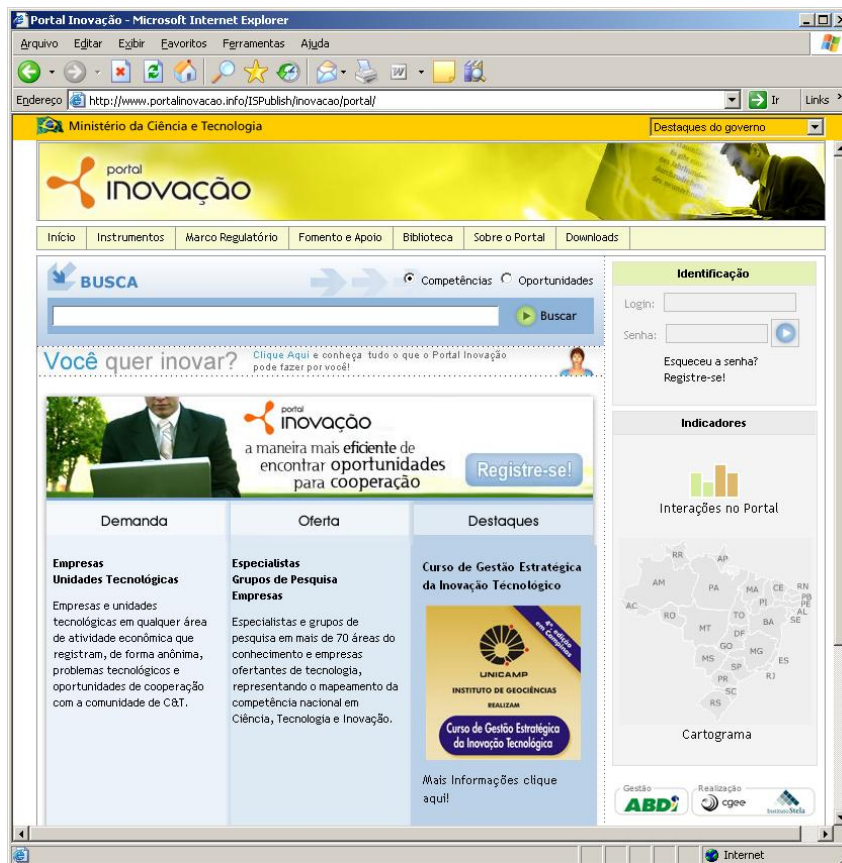


Figura 36 – Portal Inovação

Para os profissionais especialistas é necessário possuir o Currículo Lattes e tê-lo transmitido para o site do CNPq. O processo de cadastro é bastante simples; basta acessar a área de registro do portal e informar os dados pessoais que o próprio sistema do portal busca o currículo no CNPq e disponibiliza o currículo para as consultas.

Neste site, é utilizado o termo competências como metáfora quando do processo de buscas. Como são armazenados elementos curriculares, o portal procura avaliar os conhecimentos existentes intrínsecos aos currículos com base em buscas por palavras-chave. Porém, nas áreas tecnológicas as dimensões de conhecimentos possuem sinergias muito fortes com as outras dimensões, possivelmente foi feita uma analogia com a palavra competência, porém, esta competência utilizada pelo site não está nos conceitos discutidos anteriormente nesta dissertação de mestrado.

Para a busca por competências no portal, conforme apresenta a Figura 37, digita-se a competência desejada sendo que, no exemplo representado pela figura abaixo, foi efetuada a busca por “Solid State Fermentation”, e solicitado ao portal que efetue a busca.

Nesta simulação, o portal encontrou 237 especialistas em todo o país, sendo que a lista nacional é encabeçada pelo Prof. Dr. Carlos Ricardo Soccol da UFPR, onde o mesmo possui a pontuação máxima no tocante à competência buscada, demonstrada em borlas ao lado do nome do especialista e a instituição onde está vinculado. Além dos especialistas, a competência também é encontrada em 147 grupos de pesquisa. O portal permite ainda refinar a busca por estado, titulação e por área da titulação, dentre outras funcionalidades interessantes, como permitir a quem está realizando a busca visualizar o perfil curricular especialista procurado ou até mesmo estabelecer contato com este profissional diretamente através do portal.

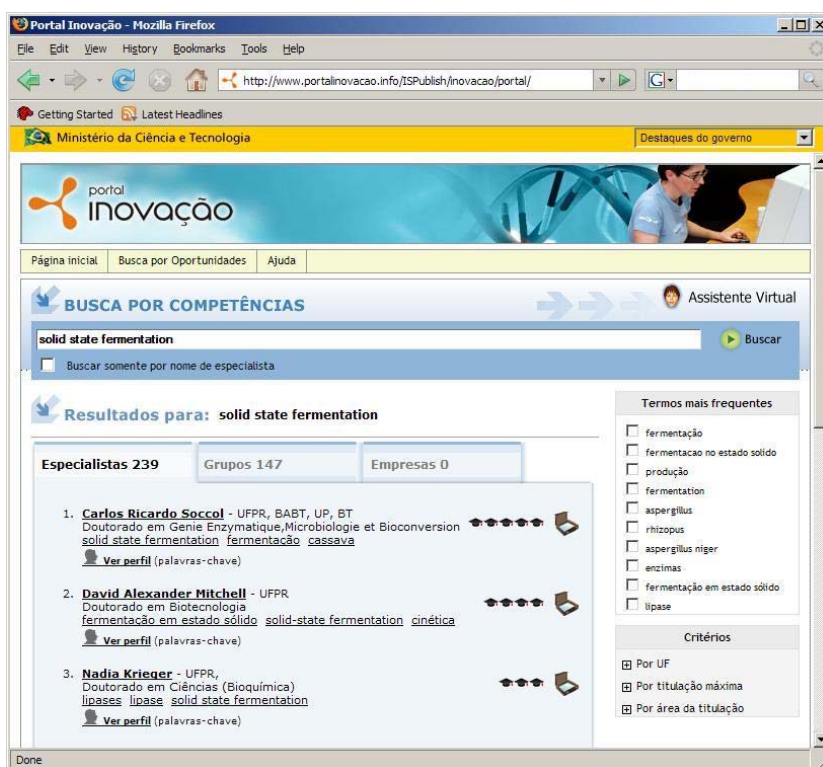


Figura 37 – Busca por competências no Portal Inovação

Em se tratando da construção do portal, foi realizada pesquisa em meios secundários na tentativa de determinar quais as tecnologias que foram utilizadas. Foram verificadas a linguagem de programação e o banco de dados utilizados para construção dos Sistemas do Instituto Stela, disponibilizados pelo CNPq através de um portal na rede Internet.

Porém, não foram encontrados detalhes de caráter técnico que pudessem vir a fundamentar a forma através da qual esta ferramenta foi construída. Então, buscou-se, nos próprios sistemas, indícios que pudessem demonstrar as tecnologias sob o qual o mesmo foi desenvolvido. De início, tentou-se verificar qual é a tecnologia em banco de dados utilizada para gerenciar os dados apresentados no site do currículo. Conforme demonstra a Figura 38, pode-se observar que os dados são recuperados através de uma conexão onde o serviço configurado é chamado “plsql1”²⁵, o que leva a crer que o Sistema Gerenciador de Banco de Dados (SGBD) utilizado é o Oracle, produzido pela empresa de mesmo nome. Este SGBD é largamente utilizado em aplicações que necessitam gerenciar grandes volumes de informações.

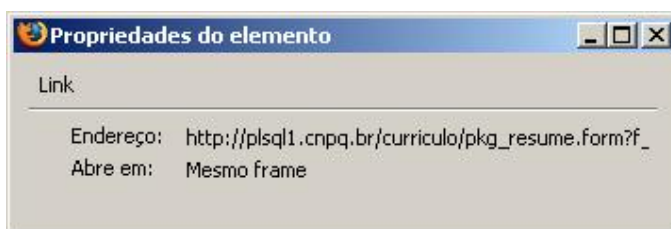
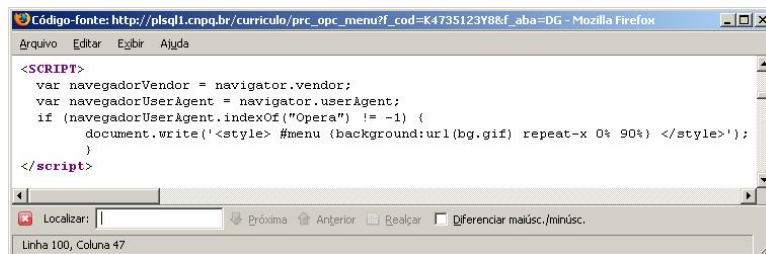


Figura 38 – Canal PLSQL do Currículo Lattes

Quanto a interface dos sistemas do Instituto Stela, a tecnologia utilizada para efetuar o gerenciamento dos *menus* do sistema é o Javascript, demonstrado pela Figura 39, largamente utilizado como ferramenta de apoio na construção de portais para a rede Internet. O restante da tela é desenhada através de um mix de tecnologias como html na construção de páginas, estilos CTCSS para formatação da interface, Java na tela e demais componentes gráficos, dentre outros.

²⁵ A linguagem PL/SQL (*Procedural Language/Structured Query Language*) é a uma linguagem procedural produzida pela própria Oracle, para facilitar o manuseio de dados em seus sistemas gerenciadores de bancos de dados.



```

Código-fonte: http://p1sql1.cnpq.br/curriculo/prc_opc_menu?f_cod=K4735123Y8&f_aba=DG - Mozilla Firefox
Arquivo  Editar  Exibir  Ajuda

<SCRIPT>
var navegadorVendor = navigator.vendor;
var navegadorUserAgent = navigator.userAgent;
if (navegadorUserAgent.indexOf("Opera") != -1) {
document.write('<style> #menu (background:url(bg.gif) repeat-x 0% 90%) </style>');
}
</script>

Localizar: | Próxima Anterior Realçar Diferenciar maiúsc./minúsc.
Linha 100, Coluna 47

```

Figura 39 – Javascript no Currículo Lattes

2.3.4 Considerações sobre os sistemas existentes de auxílio à Gestão de Competências

Neste capítulo, foram analisados os sistemas Efix produzido e distribuído pela empresa de mesmo nome, o sistema Athena desenvolvido e utilizado pela empresa Siemens e os sistemas componentes do pacote Sistemas Lattes, produzidos pelo Instituto Stela e utilizados principalmente pelas instituições de ensino e pesquisa brasileiras. Foram escolhidos estes sistemas devido a características tais como seu reconhecimento pelo mercado e a possibilidade apresentada para gestão de informações relativas a competências no auxílio ao gestor. O objetivo desta análise, é o de verificar o método através do qual os sistemas citados efetuam gestão de competências, bem como avaliar a possibilidade de se utilizar alguma das opções como ferramenta de coleta de dados para o ensaio do sistema proposto.

A empresa Siemens foi escolhida por ser considerada empresa de grande porte e devido à abertura proporcionada para a pesquisa. Destaca-se como fator determinante para a escolha o fato de possuir um sistema de gestão de competência apoiado por metodologia própria já em uso há alguns anos. Quanto ao sistema Efix, este também foi analisado pelo fato de grandes empresas o utilizarem comercialmente, sendo que, o mesmo se propõe a efetuar processos de análise e gestão de competências na organização, através de análises e validações de informações curriculares entre pares durante o processo. Os Sistemas Lattes são largamente utilizados e reconhecidos pela academia como repositório confiável de informações curriculares. A abertura proporcionada pelo Grupo Stella, desenvolvedor da ferramenta, na apresentação e demonstração dos sistemas bem como no fornecimento de materiais para este trabalho científico, foram fatores decisivos da sua inclusão nesta análise.

Neste sentido, têm-se uma dimensão do que existe em algumas das empresas da região, em função do tema que é fornecer apoio ao gestor em processos de gestão de competências. Neste contexto, verificou-se que na Siemens, o sistema foi construído para

ser utilizado apenas internamente pela empresa, sendo também estruturado sobre a metodologia SCoM (desenvolvida pela própria Siemens), o que o torna desconhecido para o mercado. No sistema Efix, como existem limitações na versão para testes disponibilizada na Internet e a empresa não demonstrou interesse em apresentar o produto de forma mais ampla, não foi possível verificar a funcionalidade plena da ferramenta nas simulações efetuadas e na documentação obtida através de meios secundários. Em se tratando dos Sistemas Lattes, em especial do Currículo Lattes, este já tem a aceitação maciça da comunidade científica, sendo que é o sistema que faz parte da tríade desenvolvida pelo Instituto Stela para o CNPq, tratados no item 2.3.3.

Diante da pesquisa efetuada, o que foi verificado que a versão disponibilizada na Internet do sistema Efix possui limitações, principalmente pelo fato de que a versão testada não possui interface para exportação de dados, além que seus dados ficam armazenados no próprio site da empresa. As informações coletadas tornam-se estanque no próprio sistema. Já o sistema Siemens é um sistema bastante evoluído, que procura ter uma visão de competências, mas seu principal limitante é que ele depende de que a pessoa declare-se no sistema e o seu superior imediato precisa verificar se ele identifica se estas competências declaradas pelo subordinado estão no contexto de certificação utilizado pelas chefias. Outro problema é que estas análises ocorrem através de processo logístico manual, onde não é apresentada uma homogeneidade de palavras, culminando em que a mesma competência seja chamada de formas diferentes pelos funcionários, visto que cada um atribui uma denominação diferente. Não existe uma matriz homogeneizada tal qual a academia possui.

A Siemens tentou montar um sistema que automatizasse o processo, mas como já mencionado neste trabalho ele está paralisado devido à reestruturação do setor de TI da empresa. Mas segundo a entrevista na empresa também foi levantado que estava havendo uma explosão de competências técnicas e isso estava tornando difícil o processo de gestão. A empresa ainda não havia efetuado um estudo para verificar se este aumento na quantidade de competências técnicas certificadas pelos chefes foi uma estratégia que aconteceu no Brasil, porque de acordo com as entrevistas ficou claro que os alemães possuem um número menor de competências técnicas trabalhadas no sistema. Desta forma, é uma ferramenta que aparentemente tem toda uma base e teoria de competências, mas hoje ela apresenta apenas as competência *soft-skill*, que se constituem de um número bem menor, mas seria necessário uma maior avaliação. Todavia, é observado que empresa precisasse de um sistema informatizado mais sofisticado para analisar sua própria projeção.

Desta forma, esta ferramenta ela apresenta limitações no contexto de gestão de competências.

O sistema Lattes IS Curriculum é um sistema que embora utilize a denominação de competências, verifica-se que simplesmente através de uma análise de densidades de ocorrência de uma determinada palavra, mas não tinha ainda uma análise de proximidade. Embora tenha sido vista uma ferramenta chamada de Portal Inovação, que já possui mecanismos que conseguem efetuar correlações, por exemplo, se informar como palavra-chave de pesquisa 1964, o sistema já faz associações com o golpe militar de 1964.

Estas correlações no software já montam uma espécie de mapa do conhecimento, que consegue analisar onde um determinado conhecimento se insere em relação aos outros. É uma estratégia bastante interessante, mas não se pode dizer que são competências porque não existe uma certificação, e sim todo o trabalho é realizado em cima do conhecimento. Porém pode-se dizer que na área acadêmica, pode-se intuir que há uma relação, uma grande probabilidade de se analisar o conhecimento explicitado e se obter uma visão das competências da pessoa, porque ela deve ter uma habilidade particular onde se apresentam os elementos que permitem que ela domine aquele conhecimento e ela teve que tomar uma atitude onde ela centrou uma decisão de pesquisa, então pode-se dizer neste caso através da análise do conhecimento pode-se ter um bom indício de uma competência, porém não existe uma certificação de pessoas que reconheçam esta competência para saber se realmente ela é competente neste determinado assunto ou não. Neste mesmo contexto, vale ressaltar que conforme citam os autores analisados neste trabalho, a competência pode necessitar de um contexto social para que ela possa “emergir”. Observa-se que com a alegação dos autores com quem publicou, pode-se na área acadêmica, evidenciando-se que isso pode não ser aplicado em uma área empresarial, onde não existe este processo de explicitação do conhecimento sustentado pelo método científico.

Neste contexto, todos os sistemas analisados carregam o nome competências, mas o que se observa é que eles não efetuam propriamente a gerência ou gestão de competências, mas buscam demonstrar conhecimentos que as pessoas avaliadas demonstram, tanto sob a visão de relacionamento social ou técnica. Neste sentido, o Quadro 4 apresentado abaixo, demonstra as principais características e dificuldades encontradas nos sistemas analisados

Quadro 4 – Relação sistemas vs. dificuldades apresentadas

Sistema	Características	Dificuldades
Currículo Lattes	Sistemas institucionalizado, formalizado onde as pessoas por questão conjuntural inserem seus dados curriculares. Exporta seu dados nativamente no formato XML.	Pessoas que estão fora da academia não sabem preencher os dados requisitados pelo sistema, ou preenchem de forma errada.
SCom/Athena	Sistema informatizado, proprietário desenvolvido pela Siemens onde as pessoas inserem suas competências e seus superiores validam sua existência.	Na coleta de dados, visto que dadas as pessoas, muitos julgam como sendo dificuldade, a correta parametrização das competências dos funcionários, uma vez que uma mesma competência quase sempre recebe várias denominações.
Efix	Sistema informatizado proprietário, desenvolvido por empresa de mesmo nome com versão de teste completa disponível na rede Internet.	Mesma dificuldade do sistema SCom/Athena, uma vez que utiliza o mesmo princípio da avaliação em 360 graus.

Fonte: o autor

Através deste quadro, observa-se que o sistema de Currículos Lattes, que possui características interessantes e facilidades, conforme categorizadas abaixo:

- Possui a estrutura para o cadastro dos dados utilizados pela academia.
- A academia já possui domínio sobre seu funcionamento, o que facilita o processos de coleta de dados. É ferramenta quase obrigatória por pessoas que precisam de incentivos principalmente governamentais para a pesquisa.
- Possui nativamente a possibilidade de exportar seus dados no formato XML, sendo formato de base de dados interessante, por ser reconhecido como um padrão para intercâmbio de dados entre sistemas informatizados.
- Como o ensaio da ferramenta será com currículos de egressos de cursos de Pós-Graduação em Engenharia de Produção e os mesmos já possuem

seus currículos cadastrado no sistema de Currículos Lattes, sua utilização pode ser uma alternativa interessante para efetuar a coleta de informações para posterior tratamento pela ferramenta informatizada baseada na tecnologia agentes.

2.4 TECNOLOGIA AGENTES NA CONSTRUÇÃO DE SOFTWARE

O objetivo desta seção é apresentar uma nova categoria de sistemas informatizados que contrasta com as tecnologias discutidas no item 2.2, representadas pelos sistemas cuja estrutura é composta por agentes de software autônomos.

Neste contexto, serão apresentados os principais conceitos da tecnologia em questão, sendo mantido o foco no ambiente JADE. Este ambiente será utilizado posteriormente no estudo de caso deste trabalho científico. Esta seção resultará em um quadro onde estarão demonstrados elementos que guiarão a construção do protótipo que será discutida no Capítulo 3, onde será efetuada a simulação da ferramenta.

Sistemas agentes são sistemas dinâmicos. Isto significa que as modificações estruturais obtidas em cada instante, além de serem irreversíveis, condicionam os resultados produzidos pelo comportamento individual nos momentos seguintes da trajetória evolutiva (CAMPOS, 1998, p. 07). Esta modalidade de construção de ferramentas de software possibilita maior flexibilidade funcional maior entre as partes do sistema, permitindo que módulos possam ser ativados conforme a demanda do próprio sistema é chamada de sistemas agentes.

Na grande maioria dos sistemas informatizados, para que o sistema comporte-se da forma desejada é necessária a intervenção de um programador. Todavia no presente momento dada a evolução tecnológica a qual estamos imersos, fazem-se necessários sistemas que tenham capacidade para adaptação e providos de mecanismos que permitam-lhes tomar decisões de acordo com os seus objetivos. Esta nova modalidade de sistemas é conhecida como sistema agente (CORCHADO, 2003, p. 101-119).

Observa-se que os sistemas agentes possibilitam o desenvolvimento de aplicações de forma independente, adicionando novos módulos que podem convergir com os antigos, para o fornecimento de serviços, viabilizando a utilização de software de qualidade sedimentado em um meio de alta tecnologia.

Tecnologias e metodologias agentes são consideradas como apropriadas para trabalhar aspectos cognitivos, inerentes à gestão de competências e comunidades ARTHUR (1994, p. 406-411).

Neste sentido, na busca de uma melhor compreensão, verifica-se a possibilidade da utilização de ferramentas informatizadas baseadas em sistemas agentes no auxílio ao gestor em situações com alto grau de complexidade.

Na seqüência, serão apresentados os conceitos de sistemas agentes e o ambiente e arquitetura do *framework* para sistemas agentes JADE.

2.4.1 Conceitos de sistemas agentes

Na visão do dicionário Aurélio, agente é uma entidade comissária ou delegada de uma pessoa, de uma instituição ou de um organismo. Nas palavras de Garcia (2005, p. 269), no que tange a sistemas informatizados (software), agentes são entidades que atuam em sistemas computacionais de acordo com regras pré-definidas de forma direta ou indireta. Para (Feber & Gasser, 1991) *apud* Garcia (2005, p. 273), um agente é uma entidade com capacidade de perceber seu ambiente, atuar nele, comunicar-se com outros agentes, oferecer serviços e apresentar determinados comportamentos frente a determinadas situações. Maes (1996) *apud* Cordeiro (2001, p. 31) define agente como um sistema computacional residente em um ambiente dinâmico e complexo, os quais têm a capacidade de perceber e atuar neste ambiente de forma autônoma, segundo objetivos pré-definidos para os quais (os agentes) foram designados. Para Yu (2000, p. 2024) agentes são entidades (semi-) autônomas que conseguem executar tarefas complexas de forma dinâmica interagindo com o seu ambiente. HASHMI (2002, p.174) define como agente, um modelo de software autônomo (ou hardware) que age de forma pro ativa, com capacidade de desenvolver de forma plena uma determinada tarefa e ao mesmo tempo respondente ao ambiente ao qual se encontra imerso. Para Garcia (2005, p. 269), os estes sistemas são caracterizados pela atuação dos agentes de forma colaborativa. Yu (2000, p. 2024) os define como um grupo de agentes autônomos buscam um objetivo.

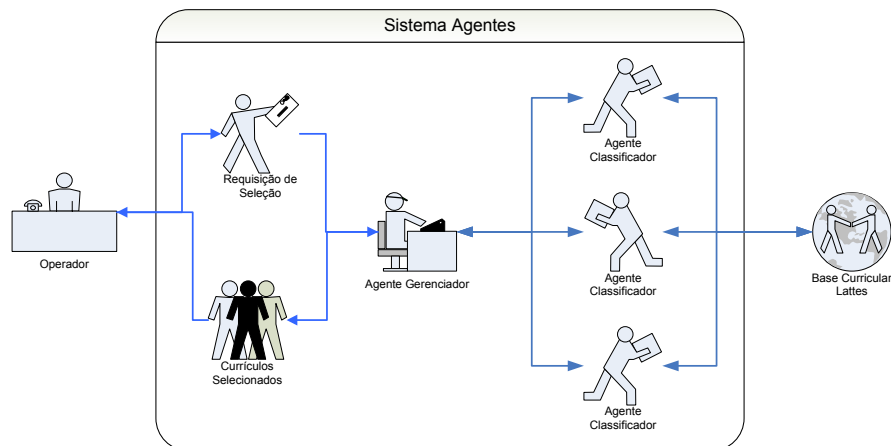


Figura 40 – Exemplo de sistema agentes

Neste sentido, o software composto por agentes é uma classe especial de sistemas informatizados que têm a possibilidade de mapear percepções em ações; sua arquitetura varia conforme a espécie e função do software, que pode ser um banco de dados, uma rede intranet ou um hardware dedicado. Sua arquitetura é responsável pela interação do agente com os seus sensores e pela execução de tarefas escolhidas por ele no ambiente através dos atuadores. Russel (2004, p.33) define como agente tudo o que pode ser considerado capaz de perceber o seu ambiente por meio de sensores e de agir sobre este por meio de atuadores, conforme demonstra a Figura 41.

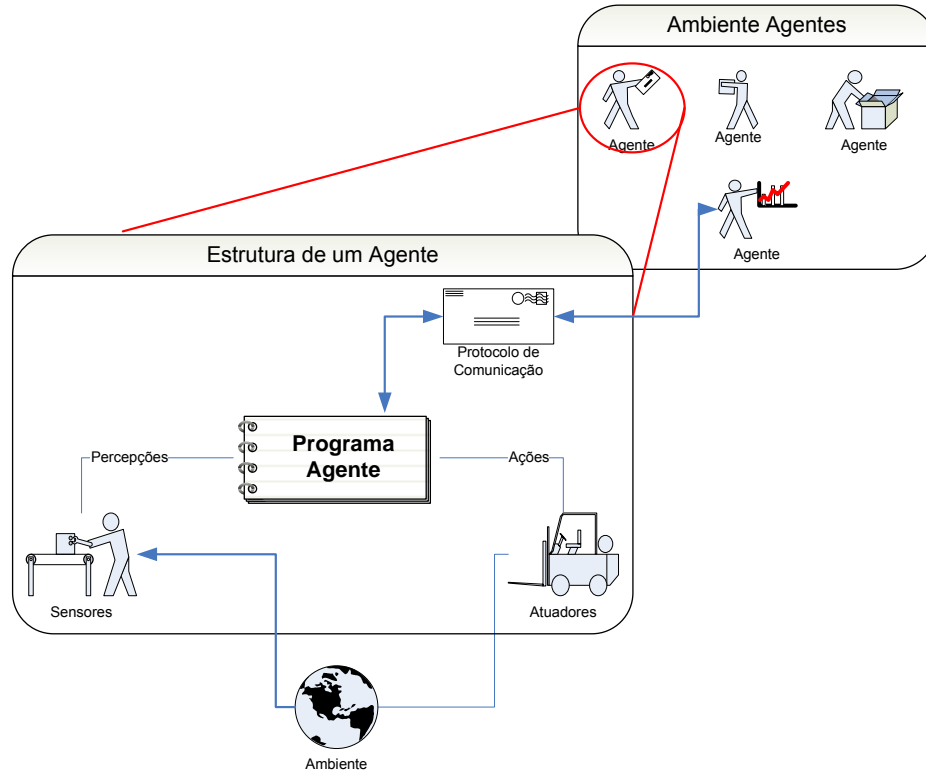


Figura 41 – Estrutura Interna de um Agente (adaptado de AOKI, 1999 p. 34)

Cordeiro (2001, p.33), Garcia (2005, p.274-275), Russel (2004, p. 33-39) evidenciam como principais características de um agente como sendo:

- Adaptabilidade (aprendizagem ou ainda autonomia): possibilidade de configurar sua atuação sem intervenções externas, harmonizando seu processo de decisão frente a novas situações;
- Comunicabilidade (ou interatividade com usuários): capacidade de se comunicar com outros agentes internos ou externos ao ambiente ao qual este está inserido.
- Reatividade: possibilidade de atuar frente à uma mudança de estado conhecida em determinado contexto.
- Mobilidade: capacidade de migrar para outros ambientes.
- Persistência: possibilidade do agente de manter um estado constante ao longo do tempo.

Buscando conceituar de uma melhor forma as características e propriedades dos sistemas baseados em agentes, Russel (2004, 46-53) fundamenta em cinco os tipos de agentes existentes, conforme descrito abaixo:

- Agente reativo simples: selecionam ações com base na compreensão atual do fato, ignorando o restante do referencial histórico de situações semelhantes vividas.
- Agente reativo baseado em modelo: agente possui um conjunto de modelos através do qual ele consegue perceber a forma que as suas ações interferem no ambiente ao qual está inserido.
- Agente baseado em objetivos: através de uma coleção de informações, o agente consegue distinguir uma série de situações desejáveis, o que permite-lhe “escolher” quais ações melhor cumprem seus objetivos (ou metas).
- Agente baseado na utilidade: um modelo de utilidade aplicado a agentes possibilita que o mesmo mapeie uma seqüência de estados, onde o mesmo consegue saber a qualidade de sua resposta frente a um estímulo.
- Agente baseado em aprendizagem: possui a capacidade de aperfeiçoar as suas ações frente a novas situações, modificando seus processos para que possam funcionar melhor no futuro; têm a possibilidade de interpretar percepções e decidir sobre ações.

2.4.2 Framework JADE – Java Agent Development Environment

O JADE é um ambiente utilizado para o desenvolvimento de sistemas agente e multi-agentes que segue o padrão definido por um instituto internacional chamado de FIPA (*Foundation for Intelligent Physical Agents*), que é uma organização define de padrões computacionais para sistemas agentes de multi-agentes na IEEE. A aderência à estes padrões, que foram criados em 1996, garante a interoperabilidade entre os sistemas heterogêneos existentes. O JADE foi aceito pelo IEEE como aderente ao padrão mundial para sistemas agentes e multi-agentes em 08 de junho de 2005 (FIPA, 2005).

Então, o *framework* JADE é uma camada de abstração de software que têm em seu núcleo um conjunto de bibliotecas que, referenciada na ferramenta de desenvolvimento

de software, permite que sejam escritos aplicativos utilizando técnicas e práticas do ambiente JADE.

Ambiente e arquitetura do JADE

Caire (2003, p. 04-06) fundamenta que o *framework* JADE, que será apresentado pela Figura 42, é composto pelos agentes e *containers* que o padrão FIPA define como sendo necessários para o seu funcionamento. Toda a Plataforma Agente JADE possui um *container* especial, chamado de Container Principal (*Main Container*), que é o primeiro *container* a ser inicializado quando a aplicação é carregada e. Ele contém dois agentes especiais e o serviço de comunicação, que irão prover funcionalidade à plataforma (CAIRE, 2003, p. 04-06) (BELLIFEMINE, 2007, p.07):

- AMS: Agente de Gerenciamento do Sistema (*Agent Management System*), ele supervisiona o acesso dos outros agentes à plataforma, sendo o responsável pelo controle e autenticação das chamadas dos agentes e o ciclo de vida dos serviços providos pela plataforma. Mantém em sua estrutura uma listagem com os identificadores dos agentes juntamente com seus estados.
- DF: Serviço de Páginas Amarelas (*Directory Facilitator*) é o serviço que provê o mecanismo de endereçamento dos agentes, aplicando-se a metáfora de “páginas amarelas” da plataforma de agente.
- Serviço de Transporte de Mensagens (*Message Transport System*) ou ainda ACC (*Agent Communication Channel*), é o componente do sistema que controla a troca de mensagens, gerenciado o uso dos protocolos de comunicação.

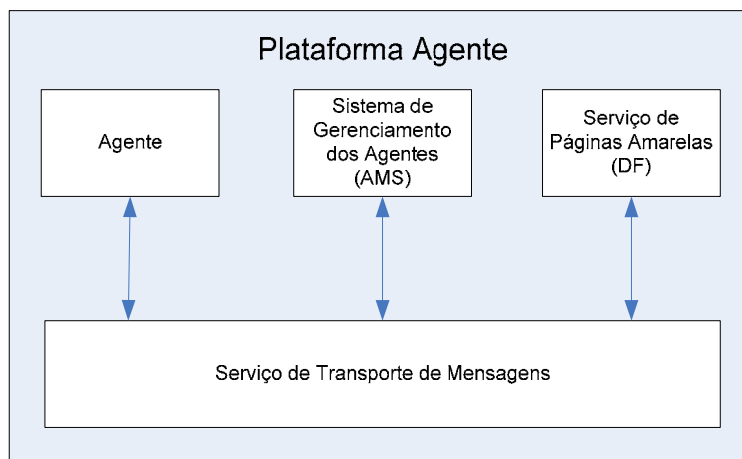


Figura 42 – Modelo estrutural básico do JADE (BELLIFEMINE, 2007, p.07)

Seguindo as especificações estabelecidas pela FIPA, os serviços de páginas amarelas e de gerenciamento de agentes comunicam-se com os agentes através de linguagem chamada FIPA-SL0. Porém, podem ser utilizadas outras implementações de linguagens de comunicação que implementem o padrão FIPA, como por exemplo o ContractNET.

Além dos elementos fundamentais necessários para o funcionamento do *framework* JADE descrito nos parágrafos acima, existe ainda a figura do elemento *Container*. Cada instância do ambiente JADE que se encontra em execução possui pelo menos um *container*, sendo que neles residem os agentes. Um conjunto de *containers* ativos é chamado de Plataforma Agente, conforme apresenta a Figura 43. Pode-se, por exemplo, em uma rede de computadores, instanciar um *container* por computador e criar agentes móveis que trafeguem por entre os computadores da rede, buscando a máquina mais interessante para se efetuar determinado processamento, transformando uma *intranet* em um “*grid* agentes”²⁶.

²⁶ Uma solução semelhante destaca-se hoje na lista de discussão de desenvolvedores da plataforma JADE na Internet, onde existe uma pesquisa em andamento buscando implementar um “grid móvel” em uma rede de telefonia celular na Europa.

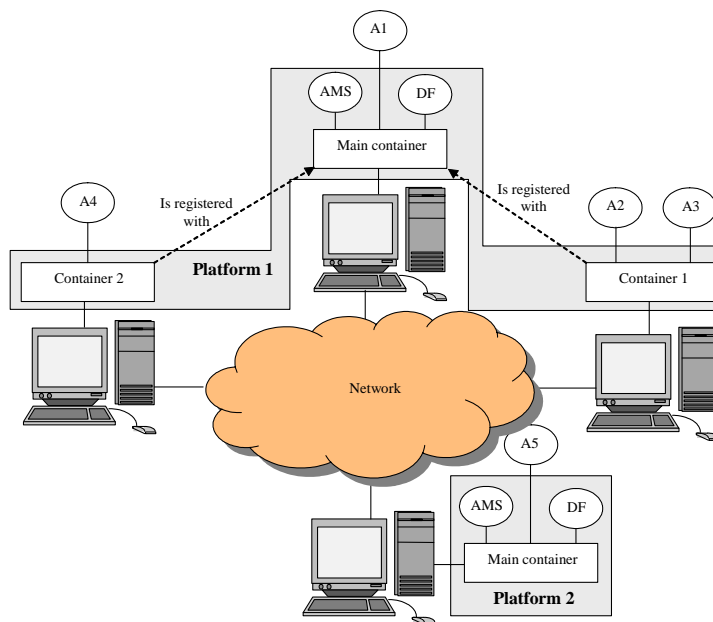


Figura 43 – Plataforma Agente e containers (CAIRE, 2003, p. 05)

Segundo Bellifemine (2007, p.09) o JADE fornece uma interface simplificada para que os agentes acessem os serviços de *AMS* e *DF*, simplesmente enviando e recebendo mensagens no formato definido pelo padrão. Como estas implementações estão completamente padronizadas pelo fato de serem muito comuns, ambas as classes podem realizar estas tarefas através de uma interface simplificada, implementada através de métodos.

A classe *Agent* no JADE, nas palavras de Bellifemine (2007, p.10-12), representa uma superclasse, o que possibilita ao programador instanciar seus próprios agentes. Isso implica em que um agente herdar as características como atributos e comportamentos de sua classe básica escrita em Java. Nele, as ações são executadas através de comportamentos, sendo estas tarefas que ocorrem de forma assíncrona e concorrente, agendados por mecanismo interno oculto ao desenvolvedor.

O agente possui um ciclo de vida bem definido, onde após iniciado podem ocorrer até quatro estados, conforme pode ser observado na Figura 44, e os estados descritos na seqüência, conforme Bellifemine (2007, p. 11). A transição entre estes estados ocorre através da chamada a métodos públicos implementados pela própria classe *Agent*, descritos na seqüência:

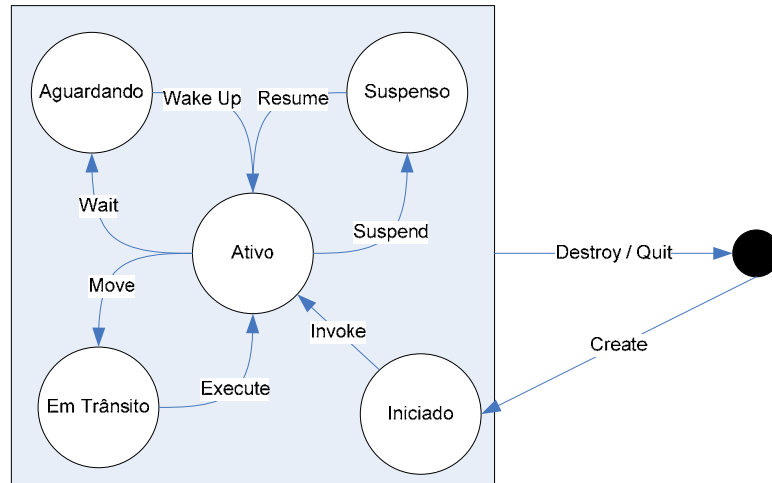


Figura 44 – Ciclo de vida de um agente (Bellifemine, 2007, p. 11)

- Iniciado: O agente foi criado, mas ainda não tem nome e nem endereçamento, o que o impossibilita de comunicar-se com outros agentes, pois não foi registrado no AMS.
- Ativo: Adquire este status após ter sido registrado no AMS. A partir deste momento é um agente válido para o ambiente e consegue acessar todas as funcionalidades disponibilizadas pelo JADE. É o único estado de onde o agente consegue acionar seus comportamentos.
- Suspensão: Sua *thread* interna está em modo suspensão e nenhum de seus comportamentos pode ser executado. O agente está parado.
- Aguardando: O agente está bloqueado, sua *thread* interna está dormente observada por um mecanismo de monitoramento interno do Java. O agente volta ao estado de ativo quando uma condição é alcançada, como por exemplo, a chegada de uma mensagem.
- Em Trânsito: Estado utilizado em agentes móveis. Um agente entra neste estado quando está migrando de um *container* para outro. O *framework* continuará colocando as mensagens em um *buffer* durante o processo e, as redirecionará após o agente estar ativo na nova localidade (*container*).
- Destruido (ou deletado): A *thread* do agente foi finalizada, seu registro foi excluído do AMS e seus comportamentos não serão mais executados.

No *framework* JADE, o agente inicia seu ciclo no momento em que seu método construtor é executado, onde o agente recebe seu identificador e se registra no AMS, definindo o seu estado como ativo. Somente após estas etapas é que o seu método de *setup()*²⁷ é executado.

De acordo com as especificações FIPA, citado por Bellifemine (2007, p. 12), o identificador de um agente possui os seguintes atributos:

- Nomeclatura única: Por padrão, a plataforma JADE cria o nome de um agente através da concatenação do nome local (definido pelo desenvolvedor), acrescido do símbolo de arroba (@), mais o nome do host onde a plataforma foi instanciada e mais a porta utilizada na comunicação (por padrão, é utilizada a porta 1099).
- Conjunto de endereços de transporte: Herdado por cada agente a partir da plataforma onde foi instanciado. É utilizado o endereço IP do computador²⁸.
- Conjunto de informadores: serviços de páginas que informa onde os agentes estão registrados, possibilitando o correto endereçamento para a troca de mensagens.

O método *setup()* é o ponto onde a atividade de qualquer aplicação estruturada na forma de agente é iniciada. Quando este método é executado, o agente já está registrado no AMS, estando na plataforma com o estado de *ativo*. Todavia, o programador ainda têm a possibilidade de utilizar-se do mecanismo de inicialização para, se necessário, modificar os

²⁷ Detalhes sobre a implementação do método *setup()* são apresentados no item 3.3.1.1.

²⁸ Houveram empecilhos para rodar a aplicação em computadores com sistema operacional Windows XP em modo *stand-alone*, durante o início do desenvolvimento da ferramenta. Após o estudo do problema, foi verificado que o referido sistema operacional possui uma funcionalidade chamada de "Microsoft Media Sensing" que, desabilita o endereço IP da placa de rede caso o cabo de rede não esteja conectado ao computador, mesmo que esteja configurado um endereço IP fixo para a referida porta, verificado através do comando *ipconfig* no prompt de comando. Para desabilitar esta funcionalidade, permitindo que a placa possua o endereço com o cabo desligado é necessário efetuar alterações no registro do sistema operacional. As instruções para efetuar o procedimento podem ser encontradas no artigo técnico "Como desabilitar o "Media Sensing" para TCP/IP no Windows", disponível no site do suporte da Microsoft em: <http://support.microsoft.com/kb/239924>

dados registrados no AMS, registrar o agente em outros *containers* ou adicionar comportamentos que serão inicializados após o término da execução do método de *setup()*. No término da execução do método de *setup()*, o JADE automaticamente direciona a execução para o primeiro comportamento da fila e troca para os próximos através de um escalonador não preemptivo do tipo “*round-robin*”²⁹. Os métodos *addBehavior()* e *removeBehavior()* podem ser utilizados para gerenciar a fila de tarefas (comportamentos) a executar (BELLIFEMINE, 2007, p.11-12).

Por sua vez, a interrupção do funcionamento do agente conforme observado em Bellifemine (2007, p.13), acontece quando o método *Agent.doDelete()* é acionado. Esta chamada pode ser executada a partir de qualquer comportamento. Em seguida, o agente entra em um estado intermediário chamado *deletado* onde o método *Agent.takeDown()* é disparado, e o agente é destruído.

A comunicação entre os agentes é efetuada através de métodos disponibilizados pela própria classe *Agent*. A troca de mensagens é executada de forma assíncrona, onde objetos oriundos da classe abstrata *ACLMessage* são transportados entre os agentes e *containers*. Os protocolos de comunicação disponíveis para a utilização no ambiente JADE encontram-se na classe *jade.proto*. O envio da mensagem pelo agente ocorre quando o método *Agent.send()* é executado.

2.4.3 Considerações sobre sistemas agentes

Seguindo o objetivo deste trabalho de pesquisa, que é desenvolver uma ferramenta informatizada baseada na tecnologia agentes para auxílio ao gestor no processo de gestão de competências. Foi efetuada esta análise do ambiente agentes JADE, com o intuito de verificar seu funcionamento e a possibilidade de sua utilização na construção do sistema informatizado.

De acordo com o estudo efetuado sobre o *framework* agentes JADE, foi verificado que o mesmo foi desenvolvido em sua totalidade no ambiente Java. Porém, existe uma versão que pode ser construída através de um compilador Java como o *ant*, por exemplo,

²⁹ Mecanismo onde os processos são dispostos em uma fila circular, onde o escalonador percorre-a, executando cada processo.

onde através de configurações de arquivos de scripts pode-se obter como resultado final uma biblioteca que pode ser inserida em aplicações .NET.

De posse desta biblioteca externa, o ambiente JADE pode ser instanciado por outras aplicações, onde podem ser criados agentes de acordo com necessidades específicas da aplicação, instanciando e sobrecarregando os métodos e propriedades escritos inicialmente para a versão Java do *framework* JADE.

A sobrecarga dos métodos permite que os comportamentos inicialmente definidos nas classes originais possam ser modificados pelo programador durante o processo de desenvolvimento, possibilitando reescrever mecanismos para adaptarem-se a necessidades e ações da ferramenta.

Neste sentido, pode-se perceber que a capacidade de adaptação, reuso e potencial de inovação que estão inerentes nos sistemas agentes, conjugados com o uso de ferramentas de desenvolvimento que possibilitem a geração de aplicações que possam ser executadas na maioria dos sistemas operacionais, possibilitam a construção de ferramentas muito mais adaptadas às necessidades atuais.

Estima-se que esta nova concepção para desenvolvimento de aplicações em software aliado ao amadurecimento de novas tecnologias poderá sedimentar bases de conhecimento e práticas, fundamentais para alimentar o processo de desenvolvimento de modelos mais eficientes e flexíveis, oferecendo aos clientes soluções mais práticas e funcionais, fazendo com que a tecnologia em questão tenha a possibilidade de agregar maior valor ao produto final desenvolvido.

2.5 CONSIDERAÇÕES SOBRE O REFERENCIAL TEÓRICO

Verificando-se a competência, existem várias teorias de competências, existem vários quadros de competências, sendo que não é o objetivo deste trabalho discutir se um quadro é melhor do que o outro em dado contexto. O objetivo desta dissertação é fazer um levantamento do que é discutido hoje na teoria, onde foi verificado que existem várias matrizes que tratam de diversas formas as diferentes dimensões de competências. Porém, no contexto do presente trabalho verifica-se que a definição que adota a competência como sendo conhecimentos, habilidades e atitudes apresenta ser consistente para este trabalho. Esta competência é adotada porque presume-se que em um ambiente científico, quando um indivíduo escolhe desenvolver seus trabalhos em determinado contexto, este precisa tomar uma decisão definindo uma linha de pesquisa, etc. isso pode ser interpretado como uma

atitude. Se o indivíduo está conseguindo escrever, ele está explicitando habilidades, então, de certa forma acredita-se que em uma análise sobre o conhecimento pode ter uma dimensão sinérgica entre os outros níveis, ou seja, em uma análise de conhecimentos, habilidades e atitudes, pode-se gerar um bom indicador da existência de determinadas competências neste contexto. Este contexto deve ser analisado de forma distinta na empresa, porque em certos casos como em uma equipe de projetos, a pessoa pode não estar explicitando suas reais competências porque não é o objetivo dela na empresa, e sim o de gerar resultados no posto de trabalho ao qual está alocada.

Percebe-se na análise do Currículos Lattes um potencial consistente, visto diante do analisado boa consistência na ferramenta, pois é bastante na comunidade científica e instituições de ensino e pesquisa nacionais. O sistema de currículo Lattes têm a possibilidade de exportar seus dados no formato XML, possibilitando que, pelo fato de já ser uma ferramenta bastante conhecida, pode-se com o seu uso garantir que os dados curriculares obtidos através do seu preenchimento sustentem a estrutura necessária para a avaliação através de outros sistemas informatizados. Neste sentido, o currículo Lattes é uma alternativa atraente para efetuar a coleta de dados curriculares. Porém, deverão ser feitos estudos mais precisos no futuro para identificar se as dimensões de habilidades, conhecimentos e atitudes estão corretamente avaliadas no que tange à forma através da qual são expressos no Currículo Lattes.

Com relação ao sistema, nas bases e linguagens estudadas verifica-se que existe convergência no mercado para uma linguagem moderna, que é o C#, sendo que pelas demais características de uso e o fato de ser um padrão aberto, se mostra interessante o seu uso. Outra linguagem que poderia ser viável seria o Java, mas por opção tecnológica e face às outras vantagens apresentadas durante a análise das linguagens e ao seu desempenho, optou-se por utilizar a linguagem C#.

Observando os sistemas discutidos que se propõe a efetuar a busca por competências, o sistema que mais se aproxima das características desejadas para efetuar a coleta de informações é o sistema Currículo Lattes do Instituto Stela. Isso se deve ao fato de que este experimenta uma sinergia interessante com este trabalho de pesquisa por já ser uma ferramenta conhecida pela comunidade científica. Embora o XML seja considerado hoje um padrão universalmente aceito para intercâmbio de dados entre sistemas informatizados, a opção por este padrão não é apenas pelas características que oferecem facilidades na implementação, mas porque os Sistemas Lattes utilizam e exportam dados para este padrão.

O ambiente JADE para o desenvolvimento dos agentes, conforme já discutido é inteiramente escrito em linguagem Java. Porém, neste trabalho de pesquisa foi identificado após a análise de diversas alternativas em linguagens de programação existentes no mercado que o C# seria a melhor opção. No futuro, acredita-se que quando o sistema estiver totalmente estruturado, devido ao fato das interrupções da máquina virtual do ambiente .NET trabalharem de forma diferente da máquina virtual Java, isso pode apresentar riscos para a estruturação de sistemas complexos com a linguagem C#. Desta forma, sabendo-se dos riscos em consideração às simulações e configurações testadas, optou-se por utilizar o JADE em ambiente C# .NET.

Diante do contexto apresentado anteriormente, observa-se que a revisão bibliográfica realizada no capítulo 02 fornece a base para a estruturação do estudo de caso do capítulo 03, que será ensaiado no capítulo 04 seguindo-se as prerrogativas apresentadas no capítulo 03 e os resultados do ensaio serão apresentados nas conclusões, no capítulo 5.

3 PROPOSTA DE SISTEMA AGENTES PARA BUSCA DE INDÍCIOS DE COMPETÊNCIAS DE EGRESSOS DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Nesta seção serão tratados os detalhes técnicos e a construção da ferramenta informatizada objetivo deste trabalho, que deverá buscar por indícios de competências em bases curriculares através do uso de sistemas agentes.

Neste contexto, serão apresentadas as tipologias e tecnologias escolhidas dentre as apresentadas e discutidas no Capítulo 2 deste trabalho. Como o objetivo é a elaboração de uma ferramenta informatizada, com base no estudo efetuado no capítulo anterior, neste capítulo será apresentada a forma através da qual foi desenvolvida a ferramenta informatizada que utiliza a tecnologia agentes.

Ao final deste capítulo, será demonstrado em um quadro, os parâmetros que irão direcionar o ensaio da ferramenta buscando validar seu funcionamento, o que ocorrerá no Capítulo 4.

3.1 TECNOLOGIAS E METODOLOGIAS ADOTADAS

De acordo com a pesquisa efetuada no referencial teórico tratado no Capítulo 2 deste trabalho, serão utilizadas as tecnologias e metodologias pesquisadas para a construção da ferramenta informatizada, conforme descrito a seguir.

Em se tratando da tipologia utilizada para a busca de indicadores de competências em bases curriculares, será utilizada a metodologia proposta por Chiavenato (2004) e apresentada no item 2.1.4.1 e discutida em 2.1.5.1. A escolha desta tipologia face às demais possibilidades apresentadas se deve ao fato do autor fundamentar que, o processo de busca por competências é regido através de comparações entre os parâmetros buscados e as características apresentadas. Desta forma, a utilização desta técnica direciona na construção da ferramenta à estruturação de processo de busca através de palavras-chave na base curricular. Isto condiciona a aplicabilidade da tipologia em relação ao escopo deste trabalho e as particularidades da ferramenta proposta.

Quanto à linguagem de programação que será utilizada na implementação deste trabalho, dentre as possibilidades tratadas no item 2.2, de acordo com as considerações tecidas no item 2.2.3, será utilizada a linguagem C# .NET por ser considerada mais

relevante dentre as demais possibilidades pesquisadas e analisadas. Esta análise foi desenvolvida através da verificação das características necessárias estabelecidas de acordo com o objetivo deste trabalho. Foi, foi escolhido dentre os ambientes de desenvolvimento existentes que suportam C#, o *Microsoft Visual C# 2005 Express Edition*, pelo fato de não incidir custos de licenciamento para sua utilização acadêmica e por já instalar toda a documentação necessária referente à linguagem de programação e bibliotecas .NET.

Quanto à base de dados, como este trabalho é um estudo que visa verificar a aplicabilidade de uma nova tecnologia de construção de software face aos modelos tradicionais, nesta primeira versão não será utilizado Sistema Gerenciador de Banco de Dados para o manuseio de dados curriculares conforme discutido nos itens 2.2.2 e 2.2.3. Então, o sistema coletor de informações curriculares escolhido irá gravar currículos em formato XML diretamente em um diretório do sistema de arquivos. A ferramenta informatizada irá efetuar a carga dos dados para efetuar as simulações deste diretório. Conforme discutido no item 2.2.3, sugere-se para utilização futura quando da implementação do repositório de informações curriculares em um Sistema Gerenciador de Banco de Dados (SGBD) a utilização do PostgreSQL. Isso se deve ao fato do mesmo ser compatível com as características necessárias do projeto e possuir a robustez necessária para suportar o crescimento exponencial de suas bases dados. O referido SGBD já está integrado ao projeto para receber as mensagens de interação dos agentes e na geração dos resultados para apresentação na tela do sistema.

No tocante ao sistema informatizado, que será utilizado na coleta de informações curriculares para a geração da base de currículos XML, de acordo com o estudo realizado no item 2.3, será utilizado o sistema de Currículos Lattes. Sua escolha se deve ao fato de permitir a compatibilidade de informações no formato XML, dentre outras características, discutidas anteriormente no item 2.3.4.

O sistema a ser construído também utilizará a tecnologia agente na construção de suas estruturas de classificação e controle, de acordo com os objetivos deste trabalho científico. Através do uso destes mecanismos chamados de sistemas agentes, tratados no item 2.4 deste trabalho, busca verificar se a tecnologia em questão é realmente adequada para o desenvolvimento de sistemas informatizados que consigam tratar de uma melhor forma aspectos cognitivos como competências e conhecimentos.

3.2 REQUISITOS DO SISTEMA

Buscando atender as premissas deste trabalho de pesquisa, foram verificados os requisitos mínimos necessários que a ferramenta informatizada precisa atender para ensaiar a tecnologia agente durante o estudo de caso. Estes requisitos demonstram a estrutura básica necessária ao funcionamento e operação do sistema informatizado (Hydra) proposto, que se resume a capacidade dos agentes efetuarem a leitura da base de currículos, classificarem estes currículos em ordem de ocorrências segundo a palavra-chave utilizada para a busca e, mostrar o resultado final ao operador do sistema. Esta estrutura pode ser observada através da Figura 45.

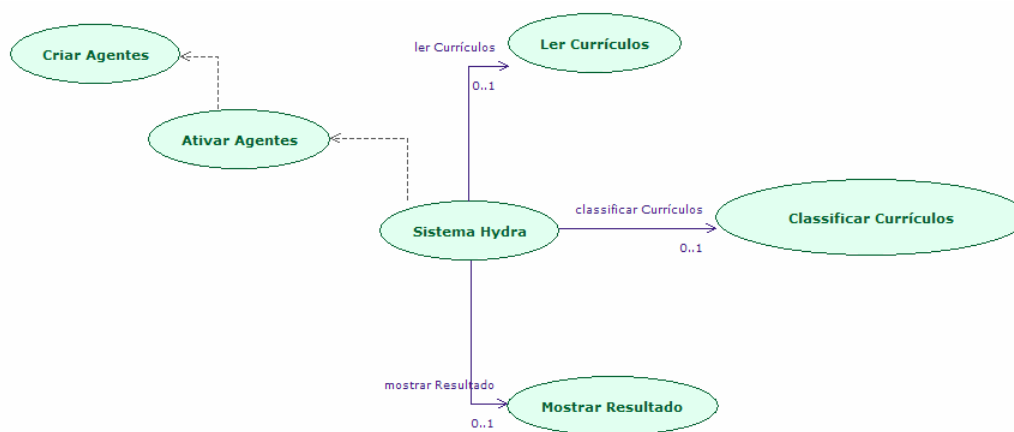


Figura 45 – Diagrama de Casos de Uso

A estruturação da ferramenta informatizada segundo propõe o diagrama de caso de uso apresentado anteriormente, resulta em um diagrama de interação de classes. Este diagrama, conhecido também por diagrama de seqüência, demonstra a forma através da qual, as classes internas do sistema operam trocando mensagens e informações. Na Figura 46, pode-se observar os agentes que compõe o sistema informatizado (AgentMasterControl, AgentListFile e AgentSearch) interagindo entre si e apresentando as informações para o operador do sistema, sendo finalizados logo em seguida.

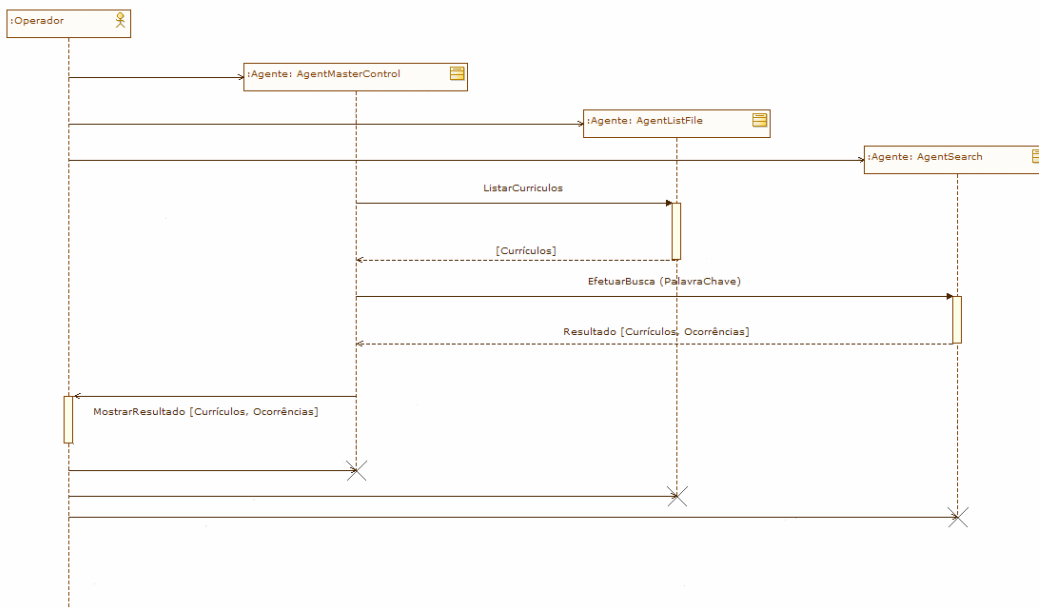


Figura 46 – Diagrama de Seqüência

Entende-se então, como requisito para a operação do sistema proposto uma estrutura informatizada composta por três agentes independentes. Estes agentes deverão ter a capacidade de trocar mensagens entre si. Um agente deverá controlar a interação dos demais e a apresentação dos resultados ao operador do sistema, sendo denominado “AgentMasterControl”. Um segundo agente irá efetuar a listagem dos arquivos para que o agente controlador possa efetuar carga de dados. Este agente será denominado “AgentListFile” e terá como premissa básica entregar ao agente controlador a listagem dos arquivos existentes que o agente precisará ler. Por fim, será necessário um terceiro agente de software que possa efetuar buscas nos dados recebidos devolvendo a resposta ao agente controlador, sendo este agente de software chamado “AgentSearch”.

3.3 CODIFICAÇÃO DO PROTÓTIPO EM LINGUAGEM C# COM JADE LEAP

Para a codificação do sistema agentes intitulado Hydra³⁰, primeiramente é necessário referenciar no projeto o ambiente .NET chamado de *JADE Leap*, que é uma biblioteca padrão DLL .NET gerada a partir de instruções contidas em Caire (2006).

Devido a problemas de compatibilidade, neste projeto a geração do DLL .NET foi executado em duas etapas distintas, onde para a geração dos arquivos java (.jar) foi utilizado o compilador java *ant* em ambiente Linux Slackware versão 12, para posteriormente em ambiente Windows XP efetuar a conversão dos referidos arquivos java para bibliotecas .NET.

Após o processo de compilação para geração do *JADE Leap*, adiciona-se a biblioteca como referência no Visual Studio C# Express Edition para que possa ser utilizada por toda a aplicação, conforme demonstra a Figura 47, onde a biblioteca aparece em destaque nas referências do projeto. Posteriormente, também será necessário inserir como referência no projeto a biblioteca *npgsql*³¹, pois é ela quem possibilita o uso do banco de dados PostgreSQL no ambiente de desenvolvimento. A biblioteca está disponível para download no site <<http://pgfoundry.org/projects/npgsql>>.

³⁰ O nome do sistema faz alusão à lendária serpente Hydra, que é uma conhecida figura da mitologia grega, destacada por possuir várias cabeças.

³¹ A biblioteca *npgsql* é desenvolvida e mantida pelo brasileiro Francisco Figueiredo Júnior, membro do projeto Mono Brasil, patrocinado pela FINEP.

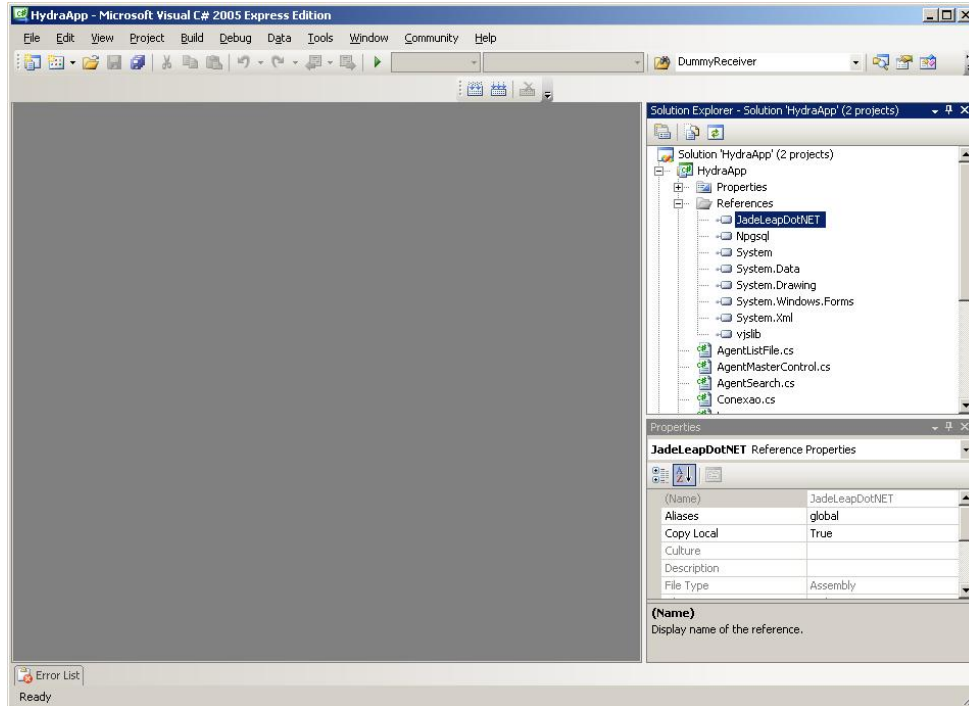


Figura 47 – JADE com Visual Studio Express

3.3.1 Agentes JADE em linguagem C#

Conforme já discutido no item 2.4.1, agentes são instâncias de software autônomas que percebem seu ambiente através de sensores e atuam quando requisitados, sendo que sua comunicação ocorre através de troca de mensagens.

Ao codificar os agentes, é necessário informar no código de qual classe do ambiente JADE que o atual agente está herdando sua estrutura base. Neste projeto, as classes utilizam a classe base *Agent*, efetuando a herança segundo diretivas do C#, demonstrado a seguir.

```
public class AgentSearch : Agent
```

Neste contexto, buscando compreender o funcionamento do sistema de uma melhor forma, sua estrutura foi dividida em três processos distintos:

3.3.1.1 Processo inicialização e encerramento dos agentes

Sempre que um agente JADE é instanciado em um *container*, pela atuação do agente *AgentController*, automaticamente é executado a sobrecarga³² do método *setup()* da classe agente, que é encarregado de efetuar a sua inicialização, conforme demonstrado pelo fragmento de código abaixo:

```
public override void setup()
```

Durante a execução deste método, são efetuadas as operações que criam a descrição do agente para logo em seguida efetuar o seu registro no *Directory Facilitator*³³. Após estes procedimentos, são inicializados os comportamentos dos agentes, através do método *addBehavior*, conforme exemplo a seguir, onde é adicionado o comportamento de recebimento de mensagens chamado *ReceiveMessageFromMControl* no agente que efetua buscas em currículos.

```
addBehaviour(new ReceiveMessageFromMControl(args));
```

Após iniciados os comportamentos, ele são executados e após a finalização do comportamento, o método que efetua a finalização do agente é executado. Da mesma forma que o *setup()*, também é necessário efetuar a sua sobrecarga, e este método é chamado de *takeDown()*, conforme exemplificado abaixo:

```
public override void takeDown()
{
    Resultado.Log("+++ AH_Search - Agente de recebimento " +
        getLocalName() + " sendo finalizado...");
    this.takeDown();
}
```

³² A sobrecarga é a aplicação de técnica de polimorfismo na classe herdada, o que permite ao desenvolvedor reconstruir um método com diferentes parâmetros.

³³ Ver item 2.4.2

3.3.1.2 Comportamentos dos agentes

Os comportamentos dos agentes são as ações que os mesmos têm a possibilidade de executar em face de um estímulo percebido no ambiente, como o recebimento de uma mensagem, por exemplo. Podem ser inicializados a partir do método *setup()* do agente conforme já discutido, mas podem ser inicializados também a qualquer momento a partir de outros comportamentos, conforme demonstra a Figura 48.

Em sua estrutura, basicamente residem duas implementações para que o comportamento execute ações. A primeira é o pelo método *action()*, que possui o código que será executado quando o comportamento for acionado. A segunda implementação é responsável pela finalização do comportamento e é implementada pelo método *done()*, que é chamado após o término da execução do comportamento, seja ele por ter alcançado um objetivo ou pela finalização de sua própria implementação pelo tipo de comportamento definido, descritos na seqüência.

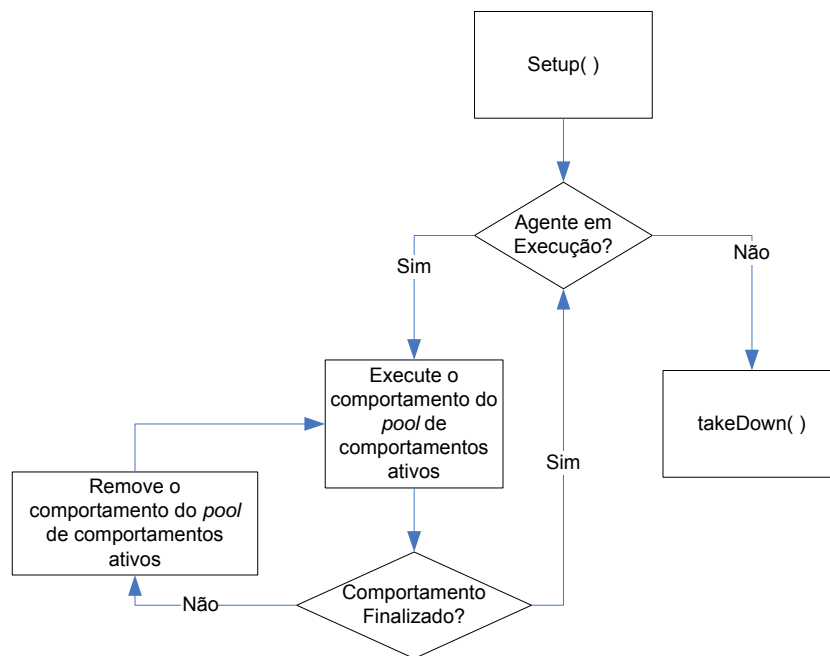


Figura 48 – Estrutura de execução do agente JADE (CAIRE, 2003, p.10)

Estes comportamentos podem ser descritos em três tipos básicos (CAIRE, 2003, p. 08):

Comportamento “One Shoot”: É o tipo mais simples implementado de um comportamento. O método *action()* do agente é executado um única vez e ao seu término,

o código é finalizado diretamente. Nele, não aparece o método *done()* como será descrito nos demais comportamentos pelo fato de que este sempre retornará o valor lógico *true* ao final de sua execução.

```
public class Nome Comportamento : OneShotBehaviour
{
    public override void action()
    {
        // Executa a ação  $\alpha$ 
    }
}
```

Comportamento “Ciclic”: Este comportamento nunca finaliza a sua execução, executando a ação sempre que é acionado. Ao contrário do tipo anterior de comportamento, este retorna sempre o valor lógico *false*.

```
public class NomeComportamento : Ciclic
{
    public override void action()
    {
        // Executa a ação  $\beta$ 
    }
}
```

Comportamento “SimpleBehavior”: É o comportamento comumente utilizado no desenvolvimento de sistemas agentes com a plataforma JADE. Têm a possibilidade de executar diferentes operações dependendo de seu estado, ou da forma que for ativado, conforme demonstra o fragmento de código abaixo. Pode-se observar a sobrecarga do método *action()* (1), e os testes das condições para execução de tarefas em (2) e (3) e a finalização do comportamento em (4) com a execução do método *done()*.

```
public class ReceiveReplies : SimpleBehaviour
{
(1)     public override void action()
        {
(2)         if (condicaoA == "valor")
            {
                // Executa a ação  $\gamma$ 
            }

            else
(3)         {
                if (condicaoB == "valor")
                {
                    // Executa a ação  $\delta$ 
                }
            }
(4)     }
}
```

```
(4) public override bool done()
    {
        return completed;
    }
}
```

3.3.1.3 Mecanismos de trocas de mensagens

As mensagens no JADE seguem os padrões e regras definidas pela FIPA, disponíveis em www.fipa.org. O mecanismo de troca de mensagens é executado de forma assíncrona entre os agentes. Cada agente possui uma espécie de “caixa postal” onde as mensagens são recebidas e a cada recebimento de mensagem o agente é notificado, conforme apresenta a Figura 49, que apresenta a forma através da qual é efetuada a troca de mensagens entre os agentes.

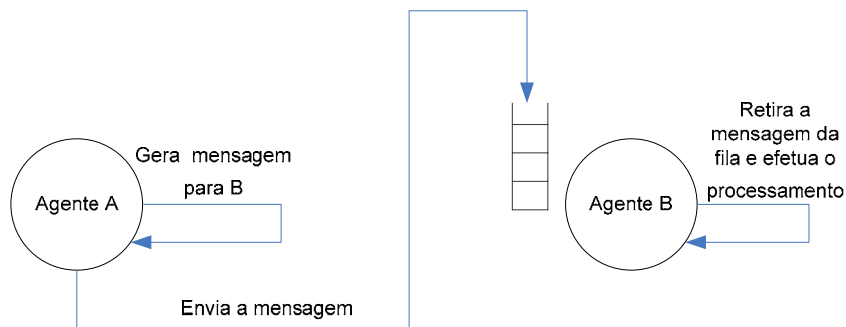


Figura 49 – Troca de mensagens entre agentes (CAIRE, 2003, p.14).

O fragmento de código demonstrado abaixo apresenta a forma através da qual uma mensagem é montada e enviada. Primeiramente, um objeto *mensagem* é criado (1) a partir da classe que possui a estrutura que permite o manuseio de objetos de mensagem. Em seguida, são inseridos na mensagem o destinatário (2), o remetente (3) e o conteúdo (4). Após, a mensagem é enviada (5) do agente para o seu destinatário.

```
(1) ACLMessage mensagem = new ACLMessage(ACLMessage.INFORM);
(2) mensagem.addReceiver(destino);
(3) mensagem.setSender(remetente);
(4) mensagem.setContent(conteudo);
(5) myAgent.send(mensagem);
```

Para o agente poder efetuar o recebimento das mensagens, também é necessário criar um método com base na classe `MessageTemplate`, conforme descrito em (1). Então é

efetuado o recebimento da mensagem em (2). No caso do sistema desenvolvido, um dos agentes verifica quem é o remetente da mensagem para poder efetuar o tratamento dos dados conforme pode ser visto em (3).

Existe também a possibilidade de se efetuar o tratamento de dados conforme o conteúdo da mensagem que for recebida pelo agente, disparando um ou outro comportamento. Pode-se abrir o conteúdo de uma mensagem através do método `.getContent()`.

```
(1) MessageTemplate mensagemRecebida =
MessageTemplate.MatchPerformative(ACLMessage.INFORM);
(2) ACLMessage mensagem = myAgent.receive(mensagemRecebida);

(3) if (aCLMessageReplay.getSender().getLocalName() == "Remetente")
    {
        //Realiza o tratamento da mensagem
    }
```

3.3.2 Estrutura do protótipo

O sistema Hydra é composto por três agentes autônomos, que trocam mensagens entre si e efetuam o processamento de currículos XML. Estes agentes são denominados AgentMasterControl, AgentListFile e AgentSearch.

De acordo com a tipologia de sistemas agentes apresentada em 2.4.1, o sistema agentes aqui desenvolvidos classificam-se como “agentes reativos simples”, pelo fato de selecionarem suas ações com base em sua compreensão atual do fato, oriunda de estímulo do meio, descartando operações advindas de seu referencial histórico.

Na inicialização do sistema, o container³⁴ principal onde residem os agentes é instanciado. Então, um agente de controle do próprio JADE, presente no *framework* chamado *AgentController* (agente controlador, ou *MAS* citado em 2.4.2), instancia os agentes no container e os ativa pelo método `start()` da instância residente no container,

³⁴ Vide item 2.4.2 na página 79.

conforme demonstrado pelo código fonte na Figura 50. Uma vez instanciado e ativado, o agente roda seu comportamento de *setup()*³⁵.

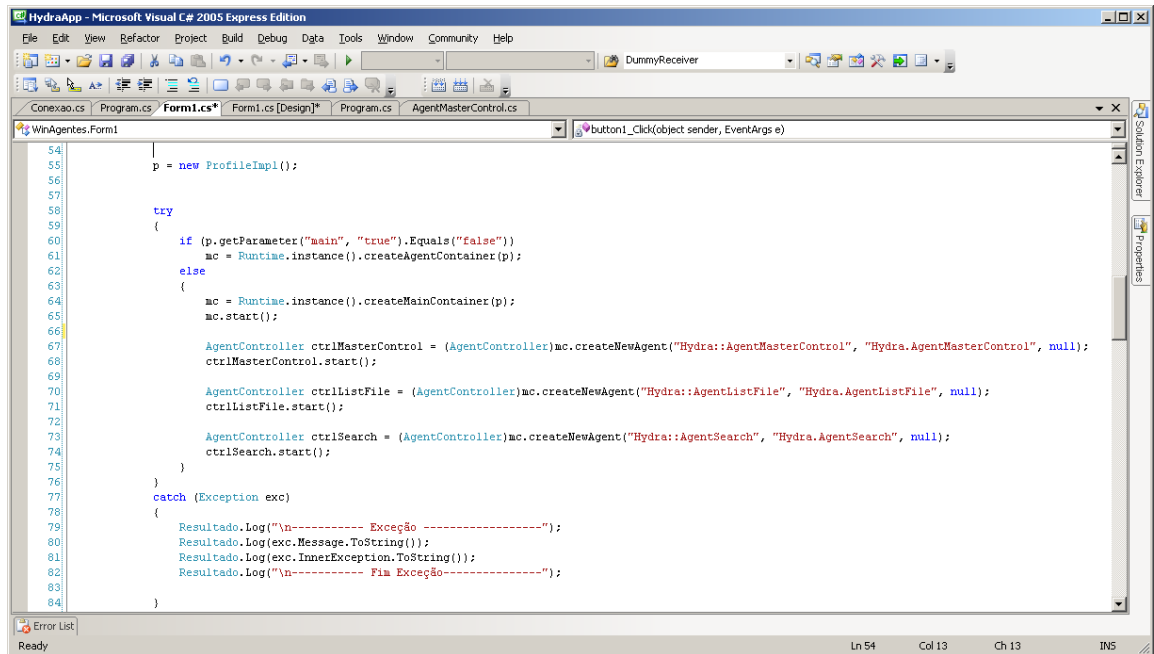


Figura 50 – Ativação dos agentes

A Figura 51 demonstra de forma resumida o funcionamento interno do Sistema Hydra. Após os agentes serem iniciados, o agente AgentMasterControl solicita ao agente AgentListCurriculum que liste os currículos estão gravados no diretório destinado à gravação de currículos XML. Este agente devolve ao agente AgentMasterControl uma lista com todos os currículos encontrados. Recebendo esta lista, o agente AgentMasterControl abre cada arquivo XML, lê o arquivo, extrai os dados das *tags* XML e efetua a sua conversão para um formato texto e os envia um de cada vez para o agente AgentSearch. Recebendo estes currículos, o agente AgentSearch efetua a busca pela palavra-chave solicitada e devolve ao agente AgentMasterControl o número de ocorrências da palavra-chave encontradas naquele currículo. Ao final do processamento de todos os currículos, o agente AgentMasterControl

³⁵ O método de *setup()* é detalhado no item 2.4.2.

gera a matriz de resultados que é mostrada na tela, contendo o caminho e nome do arquivo de currículo separados pelo caractere *underline*³⁶. Após, o processamento é finalizado.

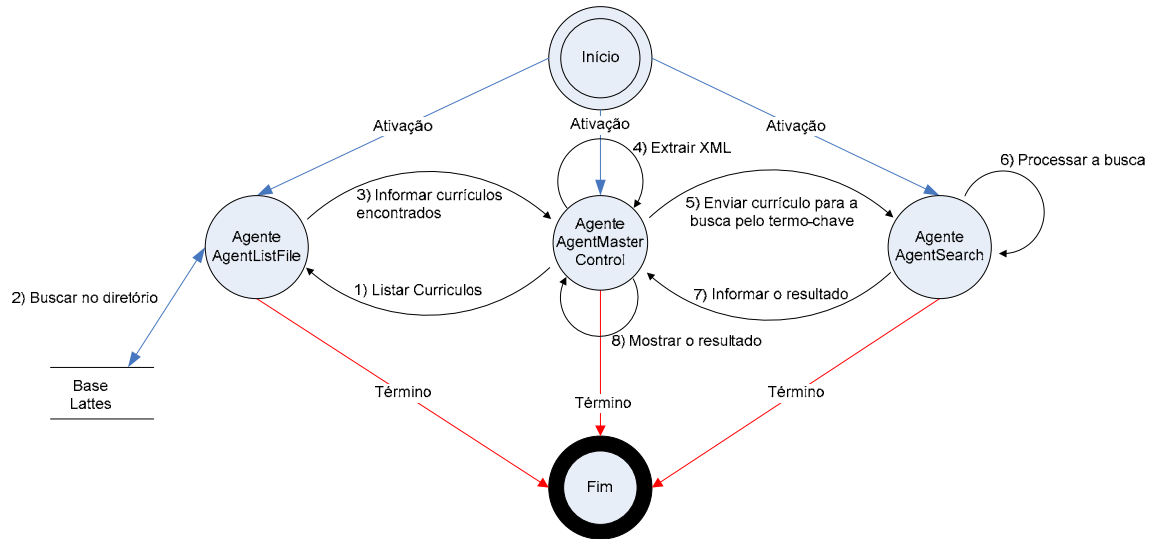


Figura 51 – Diagrama do Sistema Hydra

Na seqüência, será detalhado o funcionamento do sistema Hydra, demonstrando a forma através da qual os agentes estão interagindo através da troca de mensagens, processando e apresentando os resultados em tela.

O processamento do sistema é iniciado com o agente *AgentMasterControl* solicitando ao agente *AgentListFile* a listagem dos arquivos de currículos que serão carregados, que estão em um diretório específico da aplicação. É gerado o destinatário da mensagem (1), é criada a mensagem (2), adicionado o destinatário na mensagem (3), adicionado o remetente (4), adicionado o conteúdo (5) e finalmente a mensagem é enviada (6).

```
(1) aID = new AID("Hydra::AgentListFile", AID.ISLOCALNAME);
(2) ACLMessage aCLMessageForListFile = new ACLMessage(ACLMessage.INFORM);
(3) aCLMessageForListFile.addReceiver(aID);
(4) aCLMessageForListFile.setSender(myAgent.getAID());
```

³⁶ Foi necessário efetuar a troca da barra utilizada para demonstrar estruturas de diretórios pelo *underline*, pelo fato de que no C# a barra é um caractere de escape, o que causava problemas no momento em que o componente *datagrid* consumia o *arraylist* dos resultados gerado pelo agente *AgentMasterControl*.

```
(5) aCLMessageForListFile.setContent(msgOpen + myAgent.getLocalName());
(6) myAgent.send(aCLMessageForListFile);
```

O agente *AgentListFile* recebe a mensagem de solicitação (1) e (2), testa se a mensagem recebida é do agente *AgentMasterControl* (3). Se o resultado do teste for falso, o agente continua aguardando a mensagem; caso positivo, ele dá início ao processamento (4).

```
(1) MessageTemplate messageTemplate =
MessageTemplate.MatchPerformative(ACLMessage.INFORM);
(2) ACLMessage aCLMessageFromMasterControl =
myAgent.receive(messageTemplate);
(3) if (aCLMessageFromMasterControl == null)
{
    block(500);
}
else
{
(4) Resultado.Log("*=> AH_LFile - Mensagem recebida de: " +
aCLMessageFromMasterControl.getSender().getLocalName());
```

Sendo iniciado o processamento, o agente dispara o processo que irá efetuar a leitura dos arquivos (1) encontrados no diretório da aplicação onde estão gravados os currículos e informar quantos foram encontrados (2). Para isso, o agente executa uma estrutura de laço (2) para gerar uma matriz com uma posição (3) para cada currículo.

```
(1) arquivosLattes = Directory.GetFiles("C:\\Hydra\\XMLData", "*.xml");
(2) Resultado.Log("--> AH_LFile - Currículos Lattes a carregar: "
+arquivosLattes.GetLength(nrCurriculos));
(3) for (int i = 0; i < arquivosLattes.Length; i++)
{
    (4) mensagem += arquivosLattes[i].ToString() + ";";
}
```

Logo em seguida, o agente *AgentListFile* gera uma mensagem destinada ao agente *AgentMasterControl* informado o resultado da busca no diretório. O código abaixo demonstra a mensagem gerada. Nela, é informado o destinatário (1), é criada a mensagem (2), o destinatário é adicionado (3), o remetente também é adicionado (4), bem como o conteúdo da mensagem (5) para a mensagem em seguida ser enviada (6) apresentando o resultado da operação em tela (7).

```

(1) aID = new AID("Hydra::AgentMasterControl", AID.ISLOCALNAME);
(2) ACLMessage aCLMessageReplayForMasterControl = new
ACLMessage(ACLMessage.INFORM);
(3) aCLMessageReplayForMasterControl.addReceiver(aID);
(4) aCLMessageReplayForMasterControl.setSender(myAgent.getAID());
(5) aCLMessageReplayForMasterControl.setContent(mensagem);
(6) myAgent.send(aCLMessageReplayForMasterControl);
(7) Resultado.Log("-=> AH_LFile - Listagem de currículos enviados para " +
aID.getLocalName());

```

Ao receber a mensagem (1), o agente *AgentMasterControl* irá testar (3) quem está enviando a mensagem. Se for um reply do agente *AgentListFile*, o agente tratará a mensagem e dará início ao processo de conversão de currículos no momento em que informa o recebimento da mensagem aguardada (4).

```

(1) ACLMessage aCLMessageReplay = myAgent.receive(messageTemplate);
(2) if (aCLMessageReplay == null)
{
    block(500);
}
else
{
    (3) if (aCLMessageReplay.getSender().getLocalName() ==
"Hydra::AgentListFile")
    {
        (4) Resultado.Log("*=> AH_MControl - Mensagem de reply recebida
de: " + aCLMessageReplay.getSender().getLocalName());
    }
}

```

Recebida a mensagem com os nomes dos currículos, o agente *AgentMasterControl* inicia a conversão dos arquivos do currículos com estrutura XML para um formato string, efetuando a leitura estrutural do arquivo através da utilização da classe *XmlDocument* (1), conforme demonstra o código abaixo. Após lidos todos os *nodos* do currículo, o resultado é armazenado na variável *novaMess* (2), juntamente com uma variável criada para efetuar acompanhamento na execução do sistema.

```

(1) XmlDocument doc = new XmlDocument();
doc.Load(args[i]);
XmlNodeReader read = new XmlNodeReader(doc);
int dep = -1;
string s = "";
while (read.Read())
{
    switch (read.NodeType)
    {
        case XmlNodeType.Element:
            dep++;
            for (int a = 0; a < read.AttributeCount; a++)
            {

```

```

        s += " " + read.GetAttribute(a) + " ";
    }
    break;
}
}
curricXML = s;
(2) novaMess = cod + curricXML;

```

O resultado da pesquisa será enviado do agente *AgentMasterControl* para o agente *AgentSearch* através de mensagem, onde este poderá efetuar a busca por palavra-chave no conteúdo das mensagens enviadas, que contém os currículos, agora em formato texto. A geração da mensagem acontece primeiramente gerando a identificação de destino da mensagem (1), em seguida criada a estrutura da mensagem (2). Em seguida adicionado o destinatário (3), adicionado o remetente (4), o conteúdo (5) e a mensagem finalmente é enviada (6).

```

(1) aID = new AID("Hydra:AgentSearch", AID.ISLOCALNAME);
(2) ACLMessage aCLMessageForSearch = new ACLMessage(ACLMessage.INFORM);
(3) aCLMessageForSearch.addReceiver(aID);
(4) aCLMessageForSearch.setSender(myAgent.getAID());
(5) aCLMessageForSearch.setContent(novaMess);
(6) myAgent.send(aCLMessageForSearch);

```

Ao receber a mensagem do agente *AgentMasterControl*, o agente *AgentSearch* faz a busca pela palavra-chave fornecida utilizando um método externo³⁷ (1) ao agente, que efetua a segregação de todas as preposições, contrações e pronomes e cria uma matriz com as demais ocorrências, devolvendo apenas o resultado da contabilização do termo escolhido (2) e (3).

```

(1) WordCount c = new WordCount(curriculo, "");
(2) int i = c.count(this.SearchKeyword);
(3) string reply = nomeCurriculo + ":" + this.SearchKeyword + ":" + i;

```

Após a pesquisa, o resultado é devolvido em forma de mensagem para o agente *AgentMasterControl* conforme demonstra o código abaixo, onde é informado o destinatário

³⁷ Este módulo foi desenvolvido por Alessandro de Oliveira Binhara juntamente com o autor deste trabalho, quando do grupo de pesquisa GEPROTI da UTFPR, sob a supervisão do prof. Dr. Dario Eduardo Amaral Dergint entre 2005 e 2006. Foi mantido fora do agente por uma questão de implementação, com o intuito de se verificar a funcionalidade de componentes externos aos agentes na da aplicação JADE.

(1), a mensagem é gerada (2), adicionado o destinatário (3), o remetente (4), o conteúdo que é o resultado da pesquisa por palavra-chave (5) e é efetuado o envio em (6).

```
(1) aID = new AID("Hydra:AgentMasterControl", AID.ISLOCALNAME);
(2) ACLMessage aCLMessageReplayForMasterControl = new
ACLMessage(ACLMessage.INFORM);
(3) aCLMessageReplayForMasterControl.addReceiver(aID);
(4) aCLMessageReplayForMasterControl.setSender(myAgent.getAID());
(5) aCLMessageReplayForMasterControl.setContent(reply);
(6) myAgent.send(aCLMessageReplayForMasterControl);
```

Ao receber a mensagem, o agente *AgentMasterControl*, na forma de string (1) separado por dois pontos (:), contendo a o currículo onde foi efetuado a busca, a palavra buscada e o número de ocorrências encontrada. O próprio agente ao receber estas informações via mensagem, efetua o tratamento (2) (3) (4) e separa os valores para serem apresentados em tela (5) e (6) e o processamento dos agentes é concluído.

```
(1) string[] creply = aCLMessageReplay.getContent().Split(':');
(2) string SearchWord = creply[0];
(3) int cod = Convert.ToInt32(creply[0]);
(4) this.resultados.Add(new Resultado(arqs[cod], creply[1], creply[2]));
    if (this.nCurriculos == resultados.Count)
    {
        Resultado.Log("--> FIM DE PROCESSAMENTO DOS AGENTES...");
        foreach (Resultado tmp in resultados)
        {
            (5) ResultInfo res = new ResultInfo(tmp.Curriculo,
            tmp.Quantidade, tmp.Palavra);
            (6) res.inserir();
        }
    }
}
```

CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO

Observado o objetivo deste trabalho científico, que é o de construir uma ferramenta informatizada que possibilite auxílio ao gestor durante processos de gestão de competências, a referida ferramenta foi desenvolvida de acordo com a teoria discutida no item 2.

A linguagem de programação C# no ambiente de programação Visual C# Express 2005 foi uma escolha bastante expressiva frente as demais. O ambiente proporciona diversas ferramentas que auxiliam o processo de desenvolvimento. Dentre elas, pode-se citar o mecanismo de consulta a documentação da linguagem, a ferramenta de manuseio

dos diversos arquivos que compõe o projeto chamada de *Solution Explorer*, o navegador de objetos *Object Browser*, que facilita a escolha de objetos durante a construção do software, dentre outras.

O *framework* agentes JADE convertido em biblioteca foi integrado como referência ao projeto construído a partir do Visual Studio Express 2005. Após sua inserção, cada classe componente do projeto passou a disponibilizar as funcionalidades do *framework* JADE de acordo com a inclusão das cláusulas de *using* para a funcionalidade necessária, conforme pode ser observado no fragmento de código abaixo, onde são declaradas classes JADE além de classes nativas do .NET para serem consumidas pela aplicação.

```
using System;
using System.IO;
using System.Windows.Forms;
using System.Xml;
using jade.core;
using jade.core.behaviours;
```

A codificação e operacionalização dos agentes na linguagem C# foi o ponto crítico da construção do sistema. Como o *framework* JADE é totalmente construído na linguagem Java, toda a documentação também conduz sua utilização para a mesma linguagem. O fato de não existir documentação que demonstrasse o funcionamento de agentes JADE nativamente na linguagem C# ocasionou dificuldades durante o desenvolvimento. Em alguns momentos foi necessário recorrer ao código fonte do *framework* JADE, verificar algumas funcionalidades de forma atômica no ambiente Java e retornar ao ambiente de desenvolvimento deste projeto, buscando classes que disponibilizassem funcionalidades similares, disponíveis apenas no ambiente original. Da mesma forma na codificação, também houveram dificuldades durante o processo de depuração da aplicação. Como a ferramenta é composta por agentes que executam suas tarefas de forma autônoma, quando é inserido no código fonte da aplicação um ponto de parada (ou *breakpoint*) através do ambiente de desenvolvimento, outros agentes estão rodando em paralelo de forma assíncrona. Este fato pode conduzir o processo de execução das linhas de código do sistema para outros agentes em outros pontos da aplicação, fazendo com que *threads* de alguns processos acabem entrando em processo de “*time out*”. Isto pode ocorrer devido a um agente não repassar uma mensagem durante o intervalo de tempo esperado para outro agente. No mesmo sentido, verificou-se que o tratamento de *threads* do .NET entra em colisão com o ambiente de gerencia de *threads* do JADE. Esta colisão implica em sérias limitações, pois impedem de controlar a execução dos agentes a partir da interface gráfica,

bem como suspender determinado agente para coleta de dados de uma variável externa ou um campo de tela, tendo que ser adaptados mecanismos externos conforme descrito na seqüência. Pode-se citar como exemplo o mecanismo construído para popular os *grids* presentes na interface gráfica do sistema, que demonstram os resultados e a comunicação entre os agentes. Se estes componentes forem populados com dados diretamente, causavam o travamento do sistema. Este fato ocorria porque agentes tentavam inserir dados de forma assíncrona e muitas vezes simultânea no componente. Este problema foi contornando com o uso de um sistema gerenciador de bancos de dados, no caso o PostgreSQL. O mecanismo de integração foi construído através da biblioteca *npgsql*³⁸, juntamente com duas classes de apoio. Os agentes passaram a inserir os dados de forma assíncrona no banco de dados, de onde as informações são apresentadas na interface do sistema.

Na versão atual do sistema, a cada reinicialização da ferramenta as tabelas do banco de dados são esvaziadas. Em versões futuras, pode-se customizar a ferramenta para que ela salve o resultado das simulações já efetuadas para consultas posteriores.

Também foi necessário modificar o caminho apresentado juntamente com o nome do arquivo, na coluna NomeArquivo do *grid* de apresentação dos resultados. Este procedimento foi necessário porque no C#, a barra invertida é um caractere de escape, o que causava problemas no momento em que o componente era preenchido com dados. Então foi codificado um mecanismo que substitui a barra invertida por um caractere de sublinhado(_). Então, o caminho "C:\Hydra\XMLData\1.xml" é apresentado na tela como "C:_Hydra_XMLData_1.xml".

³⁸ A biblioteca *npgsql* é apresentada no item 3.2.

4 ENSAIO DO PROTÓTIPO AGENTES JUNTO AO PPGEF

Neste capítulo será apresentado o ensaio efetuado com o sistema desenvolvido embasado na plataforma agentes através do *framework* JADE com o ambiente de desenvolvimento C# .NET. Este ensaio têm por objetivo verificar o comportamento da aplicação desenvolvida, face a heterogeneidade de tecnologias utilizada na construção.

4.1 O PROCESSO DE ENSAIO

Após a conclusão dos trabalhos de desenvolvimento e depuração da ferramenta informatizada sedimentada em agentes autônomos de software, o ensaio do sistema foi efetuado utilizando-se como insumo uma base de currículos Lattes, salvos no padrão XML. Estes currículos foram obtidos por solicitação junto a alunos e egressos do Programa de Pós-Graduação em Engenharia de Produção (PPGEP), da Universidade Tecnológica Federal do Paraná (UTFPR), campus de Ponta Grossa, Estado do Paraná. Recebidos os arquivos, foi efetuada a tabulação de alguns termos chave, escolhidos para verificar se a ferramenta estava efetuando a busca de forma correta.

É importante ressaltar a importância da escolha do termo a ser utilizado na busca. Por exemplo, se são buscados indícios que procurem demonstrar competências em “Engenharia de Produção”, deve-se efetuar a busca por “produção”, “produto”, “manutenção”, etc.; termos diretamente ligados à competência desejada na grande área em questão.

Então, adquiridos e tabulados os currículos Lattes, os mesmos foram salvos no computador no diretório da aplicação³⁹ de onde são carregados. Neste ensaio, foram observados nos currículos quatro ocorrências distintas, onde buscam-se efetuar correlações com as grandes áreas de Engenharia de Software e Engenharia de Produção, sendo:

- Qualidade
- PostgreSQL

³⁹ O diretório em questão é o c:\hydra\xml\data. Por questão de padronização, os arquivos foram nomeados com os números de 1 a 9, desta forma: 1.xml, 2.xml, ..., 9.xml.

- Produção
- Produto

Com o intuito de facilitar a operação da ferramenta para o gestor, foi acrescentada na interface um componente do tipo *ComboBox*, com o intuito de filtrar o resultado apresentado no *grid*. Esta funcionalidade possibilita que sejam realizadas várias consultas seguidas e posteriormente, visualizados os resultados um a um, conforme apresenta a Figura 52.

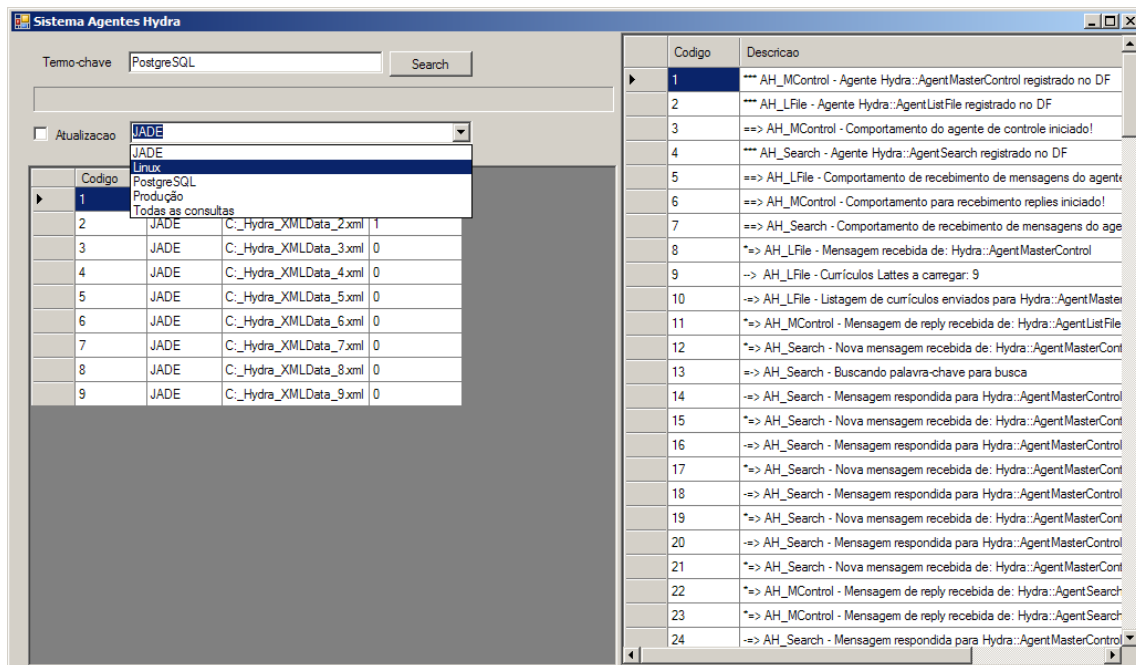


Figura 52 – Armazenamento das buscas no componente

Durante o ensaio da ferramenta, no momento em que foi efetuada a busca pelo termo chave que traz à tona indícios de algum tipo de conhecimento na área de qualidade, foram encontrados indícios em todos os currículos pesquisados, sendo que o arquivo 9.xml quantificou 43 indícios de conhecimentos neste termo buscado, conforme apresentado na da Figura 53 demonstrada abaixo.

The screenshot shows the 'Sistema Agentes Hydra' application window. At the top, there is a search bar with the text 'Palavra-chave' and a 'Search' button. Below it, a dropdown menu is set to 'Atualizacao' with the value 'Qualidade' selected. The main area is divided into two tables. The left table shows search results for the keyword 'Qualidade', and the right table shows a list of system messages or logs.

Codigo	Palavra	NomeArquivo	Ocorrencia
1	Qualidade	C:_Hydra_XMLData_1.xml	36
2	Qualidade	C:_Hydra_XMLData_2.xml	14
3	Qualidade	C:_Hydra_XMLData_3.xml	2
4	Qualidade	C:_Hydra_XMLData_4.xml	2
5	Qualidade	C:_Hydra_XMLData_5.xml	11
6	Qualidade	C:_Hydra_XMLData_6.xml	2
7	Qualidade	C:_Hydra_XMLData_7.xml	1
8	Qualidade	C:_Hydra_XMLData_8.xml	8
9	Qualidade	C:_Hydra_XMLData_9.xml	43

Codigo	Descricao
1	*** AH_LFile - Agente Hydra::AgentListFile registrado no DF
2	*** AH_MControl - Agente Hydra::AgentMasterControl registrado no DF
3	*** AH_Search - Agente Hydra::AgentSearch registrado no DF
4	==> AH_LFile - Comportamento de recebimento de mensagens do agente
5	==> AH_MControl - Comportamento do agente de controle iniciado!
6	==> AH_Search - Comportamento de recebimento de mensagens do agente
7	==> AH_MControl - Comportamento para recebimento replies iniciado!
8	*=> AH_LFile - Mensagem recebida de: Hydra::AgentMasterControl
9	--> AH_LFile - Currículos Lattes a carregar: 9
10	-=> AH_LFile - Listagem de currículos enviados para Hydra::AgentMasterControl
11	*=> AH_MControl - Mensagem de reply recebida de: Hydra::AgentListFile
12	*=> AH_Search - Nova mensagem recebida de: Hydra::AgentMasterControl
13	=> AH_Search - Buscando palavra-chave para busca
14	-=> AH_Search - Mensagem respondida para Hydra::AgentMasterControl
15	*=> AH_Search - Nova mensagem recebida de: Hydra::AgentMasterControl
16	-=> AH_Search - Mensagem respondida para Hydra::AgentMasterControl
17	*=> AH_Search - Nova mensagem recebida de: Hydra::AgentMasterControl
18	-=> AH_Search - Mensagem respondida para Hydra::AgentMasterControl
19	*=> AH_Search - Nova mensagem recebida de: Hydra::AgentMasterControl
20	-=> AH_Search - Mensagem respondida para Hydra::AgentMasterControl
21	*=> AH_Search - Nova mensagem recebida de: Hydra::AgentMasterControl
22	-=> AH_Search - Mensagem respondida para Hydra::AgentMasterControl
23	*=> AH_Search - Nova mensagem recebida de: Hydra::AgentMasterControl
24	-=> AH_Search - Mensagem respondida para Hydra::AgentMasterControl

Figura 53 – Ensaio com o termo chave "Qualidade"

Durante o processo de ensaio onde foi buscado através do termo chave na base curricular, indícios que pudessem demonstrar conhecimentos na área de manutenção, não foram encontrados indícios em todos os currículos pesquisados. Quantificaram nesta busca os currículos 1.xml com 1 ocorrência, 8.xml e 5.xml com 2 ocorrências, 2.xml e 9.xml com 3 ocorrências e o currículo 4.xml com 11 ocorrências. conforme apresentado na Figura 54.

The screenshot displays the 'Sistema Agentes Hydra' application window. At the top, there is a search bar with the text 'Palavra-chave' and a 'Search' button. Below the search bar, a dropdown menu is set to 'Manutenção'. The main area is divided into two panes. The left pane contains a table with the following data:

Codigo	Palavra	NomeArquivo	Ocorrencia
10	Manutenção	C:_Hydra_XMLData_1.xml	1
11	Manutenção	C:_Hydra_XMLData_2.xml	3
12	Manutenção	C:_Hydra_XMLData_3.xml	0
13	Manutenção	C:_Hydra_XMLData_4.xml	11
14	Manutenção	C:_Hydra_XMLData_5.xml	2
15	Manutenção	C:_Hydra_XMLData_6.xml	0
16	Manutenção	C:_Hydra_XMLData_7.xml	0
17	Manutenção	C:_Hydra_XMLData_8.xml	2
18	Manutenção	C:_Hydra_XMLData_9.xml	3

The right pane shows a list of system events with columns 'Codigo' and 'Descricao'. The events are numbered 1 through 24, with descriptions such as '*** AH_LFile - Agente Hydra::AgentListFile registrado no DF' and '==> AH_Search - Buscando palavra-chave para busca'.

Figura 54 – Ensaio com o termo chave "Manutenção"

Na Figura 55 apresentada abaixo, são demonstrados os resultados do ensaio da base curricular, onde foi buscado o termo chave "Produção". Os resultados demonstram que foi encontrado maior número de ocorrências totalizando 139 no currículo 9.xml e o currículo 3.xml apresentou o menor número, totalizando apenas duas ocorrências para o termo buscado. O resultado da simulação apresenta que todos os currículos envolvidos no ensaio desenvolveram alguma atividade relativa a produção devido às ocorrências do termo, destacando-se o currículo já citado, onde o resultado foi projetado frente aos demais, evidenciando sua participação em área tocante à produção, seja na indústria ou academia.

The screenshot displays the 'Sistema Agentes Hydra' application window. At the top, there is a search bar with the text 'Palavra-chave' and a 'Search' button. Below it, a dropdown menu shows 'Atualizacao' with 'Produção' selected. The main area is divided into two panes. The left pane contains a table with search results, and the right pane shows a list of system messages.

Codigo	Palavra	Nome/Arquivo	Ocorrencia
19	Produção	C:_Hydra_XMLData_1.xml	12
20	Produção	C:_Hydra_XMLData_2.xml	8
21	Produção	C:_Hydra_XMLData_3.xml	2
22	Produção	C:_Hydra_XMLData_4.xml	13
23	Produção	C:_Hydra_XMLData_5.xml	15
24	Produção	C:_Hydra_XMLData_6.xml	11
25	Produção	C:_Hydra_XMLData_7.xml	3
26	Produção	C:_Hydra_XMLData_8.xml	10
27	Produção	C:_Hydra_XMLData_9.xml	139

The right pane shows a list of messages with columns 'Codigo' and 'Descricao'. The messages are numbered 1 through 24 and describe various system events related to the search process.

Figura 55 – Ensaio com o termo chave "Produção"

Neste mesmo sentido, a busca pelo termo chave “Produto” retornou resultados em apenas três dos currículos que compõe a base, conforme demonstra a Figura 56. São eles o currículo 2.xml onde foram encontradas 5 ocorrências, e os currículos 8.xml e 9.xml onde foram contabilizadas apenas uma ocorrência em cada.

The screenshot shows the 'Sistema Agentes Hydra' application window. At the top, there is a search bar with the text 'Palavra-chave' and a 'Search' button. Below it, a dropdown menu labeled 'Atualizacao' has 'Produto' selected. The main area is divided into two panes. The left pane contains a table with the following data:

Codigo	Palavra	Nome/Arquivo	Ocorrencia
28	Produto	C:_Hydra_XMLData_1.xml	0
29	Produto	C:_Hydra_XMLData_2.xml	5
30	Produto	C:_Hydra_XMLData_3.xml	0
31	Produto	C:_Hydra_XMLData_4.xml	0
32	Produto	C:_Hydra_XMLData_5.xml	0
33	Produto	C:_Hydra_XMLData_6.xml	0
34	Produto	C:_Hydra_XMLData_7.xml	0
35	Produto	C:_Hydra_XMLData_8.xml	1
36	Produto	C:_Hydra_XMLData_9.xml	1

The right pane shows a list of messages with columns for 'Codigo' and 'Descricao'. The messages are numbered 1 through 24. The descriptions include agent registration, control behavior, and message exchanges related to the search for 'Produto'.

Figura 56 – Ensaio com o termo chave “Produto”

Após a apresentação do ensaio da ferramenta, o qual é concluído neste item, na seqüência serão apresentados os resultados obtidos a partir do estudo proposto neste trabalho, fazendo-se considerações sobre o desenvolvimento da ferramenta, bem como das dificuldades encontradas durante o processo.

4.2 CONSIDERAÇÕES SOBRE O ENSAIO DO SISTEMA AGENTES

O contexto deste trabalho que é o desenvolvimento de uma ferramenta fundamentada na tecnologia agentes, para auxílio ao gestor durante o processo de busca de indícios de competências para posterior mapeamento. Neste sentido, após a análise de diversos autores e ferramentas informatizadas, na busca de percepções que pudessem auxiliar na montagem do protótipo, o mesmo foi construído sedimentado nas tecnologias .Net, C#, JADE, XML e PostgreSQL.

O ensaio foi executado de duas maneiras distintas, pela execução do código-fonte através do ambiente de desenvolvimento Visual C# 2005 Express Edition, e pela execução direta do arquivo binário, construído pela ferramenta através do processo de compilação.

A diferença de se efetuar a execução do software através do ambiente de desenvolvimento versus seu binário, é que no primeiro caso o *debugger*⁴⁰ inspeciona a execução do código no *background* da aplicação, tratando eventuais situações onde ocorram falhas no código. Quando o software é executado diretamente através do seu binário, os erros são descarregados diretamente na tela do usuário como mensagem de erro do sistema operacional.

Neste trabalho, a ferramenta desenvolvida foi ensaiada de ambas as formas, tanto através do ambiente de desenvolvimento quanto pelo binário de forma isolada. Em nenhum dos casos apresentou inconsistências no código ou gerou erros de proteção no sistema operacional Windows XP.

Foi verificado através deste ensaio que, a forma utilizada na construção do sistema, delegando a coleta de dados à outra ferramenta e preocupando-se apenas com o tratamento da informação, se mostra relevante pelo fato de que é utilizada apresenta bom potencial frente as tecnologias utilizadas nas outras opções apresentadas neste trabalho. O fato de o sistema utilizar um modelo de aquisição de dados que não depende de questionários e entrevistas, possibilita à aplicação maior grau de independência. Esta independência também torna o sistema autônomo frente à processos onde em outras soluções, fazem com que gestores e subordinados validem as informações curriculares alheias em mecanismos de avaliação em 360°. Nestes mecanismos, o processo pode ser viciado pelas partes, indicando competências que não existem ou desmerecendo as reais habilidades de um colaborador por quaisquer motivos.

A arquitetura agentes mostrou-se relevante por possibilitar o desenvolvimento de células autônomas de software. Este modelo é conveniente, pois, em versões futuras pode-se estrutura a aplicação de tal forma que os agentes consigam estabelecer relações maiores entre as informações trafegadas, habilitando agentes específicos de acordo com o contexto

⁴⁰ Ou “debugador”. Software componente do ambiente de desenvolvimento que, auxilia o programador na busca por eventuais erros no código-fonte da aplicação que passam despercebidos pelo processo de compilação, manifestando-se apenas durante a execução do programa.

que está sendo tratado. Estes agentes específicos podem carregar, por exemplo, mecanismos que tratem relações de proximidade em currículos, quantifiquem determinado tipo de conhecimento via algoritmos especializados, dentre diversas outras aplicações e soluções.

5 CONSIDERAÇÕES FINAIS

Nesta seção serão apresentadas as conclusões obtidas quanto aos assuntos relacionados aos objetivos e metodologia deste trabalho, onde foram detalhadas as teorias e conceitos ligados a gestão de competências, tecnologia agentes e desenvolvimento de software.

Quanto ao objetivo de verificar os requisitos necessários para a organização de um sistema que permita apoiar as atividades de gestão de competências, pode-se concluir com base na teoria estudada que a competência é um elemento constituinte da natureza da pessoa. Para que esta competência possa manifestar-se, muitas vezes é necessária a articulação de seus conhecimentos juntamente com uma base social, sendo a competência uma propriedade emergente de conhecimentos, habilidades e atitudes. De acordo com o observado nas metodologias tratadas, este contexto de categorização de conhecimentos, habilidades e atitudes conduz à análise de competências profissionais ou técnicas, que segundo observado neste trabalho são periféricas aos demais tipos de competências e fundamentadas em conhecimentos sistematizados e tangíveis, o que as torna passíveis de avaliação. Neste mesmo contexto de competências, também foram analisadas três ferramentas existentes hoje no mercado, que se propõe a disponibilizar mecanismos para efetuar processos de gestão de competências. A primeira é chamada de Sistema Athena e foi desenvolvida e utilizada pela Siemens, estando em uso na empresa há alguns anos. A segunda ferramenta analisada é comercializada pela empresa Efix. Esta ferramenta foi incluída nesta análise devido ao fato de que grandes empresas a utilizam comercialmente. Esta solução se propõe a efetuar o processo de validação de informações curriculares entre pares, da mesma forma que o sistema Athena. O terceiro sistema analisado foi o Sistema Lattes desenvolvido pelo grupo Stella para o CNPq, composto pelo Portal Inovação, Sistema ISCurriculum e Currículo Lattes, sendo este último bastante comum no meio científico e acadêmico. Dentre estes sistemas informatizados analisados, observado-se sob a ótica de suas funcionalidades e características, a opção que se apresentou como a mais indicada para ser utilizada como coletor de informações curriculares foi o Currículo Lattes. Foi escolhida esta opção em função da análise das demais ferramentas discutidas neste trabalho. O sistema Athena se mostrou conveniente, porém é uma solução proprietária da Siemens desenvolvida para ser utilizada apenas internamente pela própria empresa. Não obstante a isso, a operação do Athena é fundamentada na metodologia SCoM, sendo necessário efetuar a avaliação entre pares, onde os pares precisam validar as competências

declaradas entre si, o que inviabilizaria o uso para este ensaio. O sistema de Currículos Lattes mostrou-se apto para este estudo por ser uma ferramenta que organiza informações comuns à academia, como linhas de pesquisa e participações em projetos, dentre muitas outras. As principais vantagens observadas são que este sistema possui exportação nativa em formato de dados XML e, que os alunos de programas de pós-graduação, foco do ensaio da ferramenta, já possuem seus dados cadastrados nesta base curricular.

Quanto ao objetivo de verificar a possibilidade da linguagem de programação C# diante das demais opções encontradas no mercado e na academia, foi observado através do estudo que as linguagens de programação mais utilizadas hoje no mercado são as orientadas a objetos. Após a análise das opções mais comuns no mercado e academia, foram obtidas como alternativas o C# .NET e o Java. O C# .NET é uma evolução dos produtos C++ e Java e, sua evolução é regida por um padrão aberto. Já o Java possui maior difusão na academia, porém a sua máquina virtual consome mais recursos de processador e memória do computador se comparado ao C# .NET. Então, para atender o escopo deste trabalho, foi escolhido o C# .NET, principalmente, pelo fato de ser um padrão aberto, além de outras características levantadas, como sua rápida difusão no mercado, facilidade encontrada em seu processo de desenvolvimento e bibliotecas nativas para manipulação de arquivos e bases de dados XML.

Quanto ao objetivo de analisar a consistência do ambiente JADE para o desenvolvimento de aplicações agentes em ambiente .NET C#, foi verificado no estudo que o ambiente em questão foi totalmente desenvolvido na linguagem Java, tanto que a linguagem leva o seu nome: *Java Agent Development Environment*. Porém, existe uma variação da biblioteca JADE que pode ser compilada através do código fonte do ambiente original para ser utilizada no ambiente C#, conforme descrito em Caire (2006). O resultado do processo é uma biblioteca .DLL padrão .NET que, pode ser inserida em um projeto e permite instanciar agentes a partir de sua classe abstrata e sobrecarregar métodos através da diretiva *override* da linguagem. Neste sentido, observou-se que havia uma grande possibilidade de adaptação do *framework* JADE ao ambiente Microsoft que, inicialmente, incentivou seu ensaio neste trabalho de pesquisa. Porém, se observou durante a construção do protótipo que há problemas de interoperação entre os dois *frameworks* (JADE e .NET), principalmente no tocante ao mecanismo de *threads* que gerencia a interface gráfica. As bibliotecas do C# utilizadas na construção de interfaces gráficas montam uma estrutura paralela para execução da janela, onde esta é executada isolada em uma *thread*. Isso implica em problemas de sincronização entre a janela de execução e as *threads* onde estão os agentes. No protótipo testado não foi possível efetuar a sincronização das *threads*

citadas. Para operacionalizar o protótipo ensaiado foram necessárias integrações com arquivos textos e sistemas gerenciadores de bancos de dados, onde foi utilizado o SGBD PostgreSQL. Também, houve dificuldades para a depuração do protótipo desenvolvido, pois como é uma solução onde os agentes de software são executados de forma autônoma e assíncrona, o ambiente de desenvolvimento *Microsoft Visual C# 2005 Express Editions* não está preparado para efetuar o tratamento deste tipo de código durante a execução em modo de depuração (*debugger*).

Quanto ao objetivo de ensaiar a ferramenta e testar suas potencialidades e limitações, foram efetuadas simulações com o intuito de verificar como o protótipo comporta-se durante a execução de processos de trocas de mensagens, buscas e apresentação de resultados. Este ensaio foi executado basicamente de duas formas distintas. Na primeira o software foi executado a partir de seu código fonte no ambiente de desenvolvimento, onde o *debugger* da ferramenta *Microsoft Visual C# 2005 Express Editions* monitorava a interação entre as partes componentes do software. A segunda forma de ensaio ocorreu através da execução do binário do software, construído durante o processo de compilação pelo ambiente de desenvolvimento. Nas duas simulações a ferramenta protótipo desenvolvida mostrou-se estável, porém quando existe uma quantia maior de currículos a serem analisados, o consumo de recursos computacionais aumenta significativamente. Os ensaios da ferramenta foram efetuados com 9 currículos Lattes, solicitados aos alunos do PPGEF da UTFPR via e-mail, onde o protótipo mostrou-se estável. Em outro ensaio, onde foram utilizados um maior número de currículos, foram carregados 194 currículos Lattes de cursos da Pós Graduação em Engenharia de Produção obtidos através da Internet⁴¹, salvos diretamente do site do Currículo Lattes um a um. Porém estes currículos mesmo possuindo a extensão .XML, não estão em formato compatível com o XML utilizado em bancos de dados para intercâmbio de informações. Então, com o intuito de viabilizar esta simulação, foi escrito um software agentes específico, que efetuava a carga do arquivo e as buscas como se fosse apenas um padrão texto comum sem nenhum tipo de filtragem para retirada das *tags* HTML e XML presentes. Durante estas simulações com maior número de currículos, a ferramenta precisava de quase 3 minutos para responder a cada requisição de busca, em um computador com processador Intel Centrino Duo e 1 Gb de memória.

⁴¹ As listagens com os nomes dos egressos foram obtidas através dos sites das universidades. Dentre elas, pode-se citar UFSCAR, UFMG, UFSC, UTFPR e UFPE.

Conclui-se então, que o uso do *framework* JADE não é viável no ambiente de desenvolvimento C# .NET, pelo fato de que o Java possui um mecanismo de gerenciamento de *threads* diferente do existente no C# .NET. Este mecanismo é essencial para a operação de agentes autônomos de software, o que resulta em *frameworks* antagônicos executando em um mesmo espaço ambiente. Quando são construídas interfaces gráficas no ambiente Java com o *framework* JADE, a própria linguagem permite que sejam construídas “interfaces gráficas agente” que conseguem comunicar-se com os agentes JADE, pelo fato de estarem em um mesmo espaço-ambiente. Então, sugere-se que, caso seja utilizado o *framework* JADE futuramente para o desenvolvimento de aplicações agentes, seja escolhida a linguagem de programação Java, que é nativa do *framework* em questão. A utilização da ferramenta Currículo Lattes como coletor de informações curriculares, para análise de alunos e egressos de cursos de Pós Graduação em Engenharia de Produção mostrou-se uma solução adequada, visto que é um aplicativo utilizado amplamente no meio acadêmico. Porém, quando analisados currículos oriundos do mercado, seria necessário estabelecer outros filtros para aquisição de dados compatíveis com outras aplicações disponíveis. Pode-se citar como exemplo os formatos suportados pelo Microsoft Word, OpenOffice e Acrobat Reader, dentre outras possibilidades.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, Lynaldo C.; ROCHA NETO, Ivan. **Ciência, Tecnologia e Regionalização**: Descentralização, inovação e tecnologias sociais. Rio de Janeiro: Editora Garamond Ltda. 2005.

ARTHUR, W. Brian. **Inductive reasoning and bounded rationality** (the El Farol problem).. American Economic Association Papers. 1994

ABEPRO Associação Brasileira de Engenharia de Produção. **Áreas da Engenharia de Produção**. Disponível em <<http://www.abepro.org.br/areas.asp>> Acesso em 20 dez. 2005

AMARAL, Roniberto M; POSSATI, Mário A.; FARIA, Leandro I.; ALLIPRADINI, Dário H; PEREIRA, Neocles A. **Uma visão da produção científica nos Anais do Encontro Nacional de Engenharia de Produção através da Bibliometria**. 2004. Artigo. XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC.

AOKI, A. R. **Uma Proposta de Arquitetura Multi-Agente para Operação de Sistemas Elétricos**. 1999. Tese de Doutorado, Universidade Federal de Itajubá, Itajubá, SP.

SANTOS NETO, WALKAMAR. Avaliação de Recursos Humanos, Utilizando o SYSMCDA, Sistema Informatizado de Suporte à Aplicação da Metodologia Multicritério de Apoio à Decisão. 2001. Dissertação. Universidade Federal de Santa Catarina, PPGEF.

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. **Fundamentos da Programação de Computadores**. São Paulo: Editora Prentice Hall. 2002.

BARBALHO, Sanderson C. M.; ROZENFELD, Henrique; AMARAL, Daniel C. **Modelando Processos de Negócio com UML**. 2002. Artigo. XXII Encontro Nacional de Engenharia de Produção. Curitiba, PR.

BELLIFEMINE, Fabio; CAIRE, Giovanni; TRUCCO, Tizianna; RIMASSA, Giovanni. **JADE Programmers Guide**. 2007. Disponível no serviço SVN do endereço <<https://avalon.csel.it/svn/JADE/trunk>>. Acesso em 30 jun. 2007.

BORSATTO, Gabriela; SHIBATA, Tatiane; SANTOS, Elibel L. **Aplicação da gestão de competências por processos**: um estudo de caso em uma empresa petrolífera. 2006. Artigo. XXVI Encontro Nacional de Engenharia de Produção. Fortaleza, CE.

CAIRE, Giovanni. **JADE Tutorial: JADE Programming for Beginners**. 2003. Disponível em: <<http://JADE.tilab.com/>> Acesso em 05 ago. 2005.

CAIRE, Giovanni, PIERI, Federico. **Leap User Guide**. 2006. Disponível em: <<http://JADE.cselt.it/doc/LEAPUserGuide.pdf>>. Acesso em 05 jul. 2007.

CAMPOS, Fred L. S.; COSTA Marcelo A. **Tecnologia e Sistema Nacional de Inovação: Uma Abordagem Complexa**. 1998. Artigo. XVIII Encontro Nac. de Eng. de Produção, Niterói, RJ.

CARLETTO, Balduir.; FRANCISCO, Antonio C.; KOVALESKI, João L. **Competências Essenciais: contribuições para o aumento da competitividade**. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

CHEU, Dwight; LINDEN, Brian. **SQL Language Reference Manual**. USA: Ed. Oracle Press. 1990.

CHIAVENATO, Idalberto. **Gestão de Pessoas: O novo papel dos recursos humanos nas organizações**. Rio de Janeiro: Editora Campus. 1999.

CHIAVENATO, Idalberto. **Gestão de Pessoas: e o novo papel dos recursos humanos nas organizações**. Rio de Janeiro: Segunda Edição. Editora Elseveier. 2004.

COLAUTO, Romualdo D.; BEUREN, Ilse M. **Proposta para avaliação da gestão do conhecimento em uma empresa comercial**. 2004. Artigo. XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC.

COOPER, Robert; SAWAF, Ayman. **Inteligência Emocional na Empresa**. Rio de Janeiro: Editora Campus. 10ª. Edição. 1997.

CORCHADO J. M.; LAZA R.; BORRAJO L.; YAÑEZ J. C.; VALIÑO M. **Increasing the Autonomy of Deliberative Agents with a Case-Based Reasoning System**. 2003. Artigo. International Journal of Computational Intelligence and Applications. Vol. 3, Nº 1.

CORDEIRO, Arildo D. **Concepção e implementação de um sistema multi-agentes para gestão da comunicação de dados “on line” entre sistemas**. 2001. Dissertação. Universidade Federal de Santa Catarina, PPGEP.

CORMEN, Thomas H., LEISERSON, Charles E., RIVEST, Ronald L. **Introduction to Algorithms**. The MIT Press. Cambridge, USA: 1998.

CURY, Antônio C. H. Formação de Competências no Ensino Superior: um estudo de caso nas instituições de ensino superior na cidade de Cascavel/PR. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

DANTAS, Anderson B.; SANTOS, Paulo F. C.; OLIVEIRA, Emmanuele S.; BARRETO, Renata R.; MILITO, Cláudia. **Pré-avaliação de um novo instrumento para a identificação do potencial empreendedor**. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

DATE, C.J. **Introdução à Sistemas de Bancos de Dados**. Rio de Janeiro: Editora Campus. 2000.

DERGINT, Dario E. A. **Apprentissage Collectif et Milieux Innovateurs: Étude de Cas à Grenoble et Simulations Multi-Agents**. Tese. 1999. UTC, Compiègne, França.

_____. **Localidade e Aprendizagem Organizacional Distribuída: Estudo de Caso da ZIRST de Grenoble**. 2001. Artigo. IV ISKM/DM - International Symposium on Knowledge Management / Document Management. Curitiba, PR.

Dicionário Aurélio Eletrônico. Século XXI. Versão 3.0. Lexikon Informática Ltda. Novembro de 1999.

DUBAR, Claude. **A sociologia do trabalho frente à qualificação e à competência**. Educ. Soc., Campinas, v. 19, n. 64, 1998. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S010173301998000300004&lng=en&nrm=iso>. Acesso em: 26 out. 2006.

DUIZITH, José L.; WIVES, Leandro K.; TAGLIASSUCHI, Gustavo; LOH, Stanley. **Apoio à gestão de competências: software para análise de conceitos em currículos**. Artigo. Simpósio Internacional de Gestão do Conhecimento IKSM . 2003.

DURAND, Thomas. **Forms of incompetence**. In: *Fourth International Conference on Competence-Based Management*. Oslo: Norwegian School of Management. 1998.

DUTRA, Joel S. **Gestão de Pessoas: modelo, processos, tendências e perspectivas**. São Paulo: Editora Atlas, 2002.

Efix. Visão Geral In: **Efix Avaliações:** 2005. Disponível em <http://www.efixtech.com/inicial.aspx?menuS=2&menuI=1&link=geral/fr_check_list.aspx> Acesso em 01 mar. 2007.

Efix: Visão Geral. In: **Monte seu projeto de avaliação de Competências.** 2005. Disponível em <http://www.efixtech.com/inicial.aspx?menuS=2&menuI=1&link=geral/fr_check_list.aspx> Acesso em 01 mar. 2007.

Efix: **Por que o Efix Avaliações?** 2005. Disponível em <http://www.efixtech.com/inicial.aspx?menuS=2&menuI=1&link=geral/fr_check_list.aspx> Acesso em 01 mar. 2007.

EL-HANI, Charbel N. **Níveis da Ciência, Níveis da Realidade:** Evitando o dilema holismo/reducionismo no ensino de ciências e biologia. Tese. Faculdade de Educação da Universidade de São Paulo. São Paulo: 2000.

FAÉ, Cristhiano S.; RIBEIRO, José L. **Um Retrato da Engenharia de Produção no Brasil.** 2004. Artigo. XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC.

FIPA. THE FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. Disponível em: <<http://www.fipa.org/>> Acesso em 14 nov. 2005.

FLEURY, Afonso C.; FLEURY, Maria T. L.; **Estratégias Empresariais e Formação de Competências..** São Paulo: Atlas. 2000

FLEURY, Maria T.; FLEURY, Afonso. **Desenvolver Competências e Gerir Conhecimentos em Diferentes Arranjos Empresariais:** O caso da indústria brasileira de plástico. p. 189-211 in: *Gestão Estratégica do Conhecimento.* São Paulo: Atlas. 2001.

FLINK, Richard J. S.; VANALLE, Rosangela M. **Gestão por competências:** um novo modelo de avaliação. 2003. Artigo. XXIII Encontro Nacional de Engenharia de Produção. Ouro Preto, MG.

_____. **O desafio da avaliação de competências.** 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

FREIRE, José E.; SEIXAS, José A.; CAZARINI, Edson W. **Competência Atual na Sociedade da Informação.** 2001. Artigo. XXI Encontro Nacional de Engenharia de Produção, Salvador. BA.

GARCIA, Ana C. B; SICHMAN, Jaime Simão. **Agentes e Multiagentes**. 2005. Artigo. IN: Sistemas Inteligentes: Fundamentos e aplicações / organização Solange Oliveira Rezende. – Barueri, SP. Editora Manole.

GIACOBBO, Elisabeth M.; Francisco, Antonio C.; PILATTI, Luiz A. **Competindo Através da Competência Essencial**. 2004. Artigo. XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC.

GIACOBBO, Elisabeth M.; SARRACINI, Noeli S; LIMA, Silvana M. V.; FRANCISCO, Antonio C. **Mapeamento de Competências: O Caso da Cia. Iguaçu de Café Solúvel**. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

GRACIOLI, Clarissa; GROHMANN, Márcia Z.; VIEIRA, Kelmara M. **Impacto do capital intelectual na performance organizacional**. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

HARZALLAH, Mounira; BERIO, Giuseppe; VERNADAT, François. **Analysis and Modeling of Individual Competencies: Toward Better Management of Human Resources**. IEEE Transactions on Systems, Man and Cybernetics – Part. A: Systems and Humans, Vol. 36, no. 1, p. 187. Janeiro de 2006.

HEUSER, Carlos A. **Projeto de Banco de Dados**. 4ª. Edição. Instituto de Informática da UFRGS. Rio Grande do Sul: Editora Sagra Luzzatto. 2001

HASHMI, Z.I.; ABIDI, S.S.R.; CHEAH, Yu. N. **An intelligent agent-based knowledge broker for enterprise-wide healthcare knowledge procurement**. Computer-Based Medical Systems, 2002. (CBMS 2002). Proceedings of the 15th IEEE Symposium on 4-7 June 2002 Page(s):173 - 178. Digital Object Identifier 10.1109/CBMS.2002.1011373

HOLZNER, Steven. **C++ Black Book**. São Paulo: Editora Makron Books. 2001.

JOHNSON, Björn; LUNDVALL, Bengt-Åke; **Promoting innovation systems as a response to the globalizing learning economy**. GEI UFRJ. Nota Técnica 4. 2000. Notas Técnicas - Fase III; Rio de Janeiro.

KORTH, Henry F.; SILBERSCHATZ, Abraham. **Sistemas de Bancos de Dados**. 2ª Edição. São Paulo: Editora Markron Books. 1995.

LAKATOS, Eva M.; MARCONI, Marina de A. **Fundamentos de metodologia Científica**. 4ª Edição. São Paulo: Editora Atlas. 2001.

LAUDON, K.; LAUDON, J. P. **Sistemas de Informação com Internet**. 4ª Edição. Rio de Janeiro: Editora LTC Livros Técnicos e Científicos Ltda. 1999.

LE BOTERF, Guy. **De la compétence: essai sur un attacteur étrange**. Paris: Les Éditions D'Organizations. 1995

LEITE, Maria S. A.; BORNIA, Antonio C.; COELHO, Christianne C. S. R. **A contribuição da teoria da complexidade à modelagem de sistemas**. 2004. Artigo. XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC.

LÉVY, P. **O que é virtual?** São Paulo: Editora 34. 1996

LIMA, Claudinete S.; URBINA, Lígia M. **Investimentos em capital humano no contexto das estratégias competitivas de Michael Porter**. 2003. Artigo. XXIII Encontro Nacional de Engenharia de Produção. Ouro Preto, MG.

LOUDON, Kyle. **Dominando algoritmos em C**. Rio de Janeiro: Editora Ciência Moderna Ltda. 2000.

LUNDEVALL, Bengt-Ake. **Políticas de inovação na era da economia do aprendizado**. Revista Parcerias Estratégicas, nº 10, março de 2001; Ministério da Ciência e Tecnologia.

MARCHEZAN, Marcos L. **Plataforma IS Curriculum**. 2006. Apresentação realizada no Instituto Stela, Florianópolis, SC em 25 mai. 2007.

MASKELL, Peter; MALMBERG, Anders. **Localised Learning and Industrial Competitiveness**. Regional Studies Association European Conference on "Regional Futures". 1995. University of Gothenburg, Sweden.

McCLELLAND, David C. **Testing for competence rather than for "intelligence"**. 1973. American Psychologist Vol. 28, Janeiro. p. 1-40. PMID: 4684069. Disponível em <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=File&DB=pubmed>. Acesso em 15 jul. 2006.

MICHILINI, Flávia V. S.; GOULART, Christiane P.; BREMER, Carlos F. **Modelagem de Dados para a Busca de Competências para Formação de Empresas Virtuais**. 1999. Artigo. XIX Encontro Nacional de Engenharia de Produção. Rio de Janeiro, RJ.

MORIN, Edgar. **Introdução ao Pensamento Complexo**. Porto Alegre: Editora Sulina. 2005

NONAKA, I.; TAKEUCHI, H. **Criação de Conhecimento na Empresa**. Rio de Janeiro: Editora Campus. 1997.

OLIVEIRA, Uálisson R.; RODRIGUEZ, Martius V. **Gestão da diversidade: além de responsabilidade social, uma estratégia competitiva**. 2004. Artigo. XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC.

PEREIRA, Luciana A.; CAVALCANTI, Marcos C. B. **Mapeamento de competências: entendendo alguns conceitos**. 2002. Artigo. Simpósio Internacional de Gestão do Conhecimento IKSM. Curitiba, PR, Brasil.

PEREIRA, Marco A. C. **Gestão por Competências: Estudo de casos em indústrias químicas no Brasil**. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

PORTAL LATTES. In: Histórico do Currículo Lattes. **Abertura e padronização XML**. Disponível em <<http://lattes.cnpq.br>>. Acesso em 05 dez. 2005.

PORTAL LATTES. In: CNPq Conselho Nacional de Pesquisa. **Diretório de Grupos de Pesquisa** Disponível em: <<http://www.cnpq.br/gpesq/historico.htm>>. Acesso em 07 abr. 2007.

PORTAL LATTES. In: CNPq Conselho Nacional de Pesquisa. **Histórico do Currículo Lattes** Disponível em: <http://lattes.cnpq.br/conheca/con_hist.htm>. Acesso em 03 mar. 2007.

RABECHINI JÚNIOR, Roque; Carvalho, Marly Monteiro. **Competências em Equipes de Projetos**. 2001. Artigo. IX Seminario Latino Iberoamericano de Gestión Tecnológica ALTEC. San José, Costa Rica.

REVISTA INFORME. Número 183. Distrito Federal: Brasília. Abr. 2007.

REIS, G. G. **Avaliação 360 graus**: um instrumento de desenvolvimento gerencial. São Paulo: Editora. Atlas, 2000.

RICH, Elaine; KNIGHT, Kevin. **Artificial Intelligence**. 1991. Ed. McGraw-Hill. International Edition. Singapore.

_____. **Inteligência Artificial**. 1993. Ed. Makron Books. São Paulo, SP.

ROCHA NETO, Ivan. **Ciência, tecnologia & inovação**: enunciados e reflexões: uma experiência de avaliação de aprendizagem. 2004. UCB/Editora Universia. Brasília, DF.

_____. **Gestão de Organizações**. São Paulo: Editora Atlas, 2003.

RESENDE, Enio. **O livro das competências**: desenvolvimento das competências: a melhor auto-ajuda para pessoas, organizações e sociedade. Rio de Janeiro: Editora Qualitymark. 2000.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. 2ª. Edição. Rio de Janeiro: Editora Elsevier. 2004.

SANTOS NETO, WALKAMAR. **Avaliação de Recursos Humanos, Utilizando o SYSMCDA, Sistema Informatizado de Suporte à Aplicação da Metodologia Multicritério de Apoio à Decisão**. 2001. Dissertação. Universidade Federal de Santa Catarina, PPGEP.

SATO, Carlos E. A. **Gestão Corporativa de Projetos para Instituições de Pesquisa Tecnológica**: Caso LACTEC. 2004. Dissertação. Universidade Tecnológica Federal do Paraná, PPGTE.

SEBESTA, Robert W. **Conceitos de Linguagem de Programação**. 4ª. Edição. Porto Alegre: Editora Bookman. 2000.

SEIFFERT, Peter Q. **Modelo de Gestão Humana para Empresas Intensivas em Capital Intelectual**: um ensaio na Embraer S. A. 2002. Tese. Universidade Federal de Santa Catarina, PPGEP.

SIEMENS Co. COM NSC AC, **Strategic Competence Management**. January, 2006.

SILVA, Marco; RACHID, Alessandra. **Gestão por Competências – Um estudo inicial**. 2003. Artigo. XXIII Encontro Nacional de Engenharia de Produção. Ouro Preto, MG.

SVEIBY, Karl E. **A nova riqueza das organizações**. Rio de Janeiro: Editora Campus. 1998.

UBEDA, Cristina L.; SANTOS, Fernando C. A. 2002. **Gestão de desempenho por competências como elemento viabilizador das estratégias de recursos humanos e de produção**. Artigo. XXII Encontro Nacional de Engenharia de Produção. Curitiba, PR.

VENTURINI, Jonas C.; MADRUGA, Lúcia R. J. G.; MINELLO, Ítalo F. **O desenvolvimento comportamental como estratégia para a busca de sintonia com os valores e propósitos organizacionais: o caso da CVI refrigerantes**. 2005. Artigo. XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS.

W3C In: **Word Wide Web Consortium**. Disponível em: <www.w3c.org>. Acesso em 20 dez. 2006.

YOURDON, Edward. **Administrando técnicas estruturadas**: estratégias para desenvolvimento de software nos anos 90. Rio de Janeiro: Editora Campus. 1993.

YU, Li; BIQING, Huang; WENHUANG, Liu; HONGMEI, Gon; CHENG, Wu. **Knowledge based decision support system for matchmaking of enterprise competence**. Systems, Man, and Cybernetics, 2000 IEEE International Conference on Volume 3, 8-11 Oct. 2000 Page(s):2023 - 2027 vol.3. Digital Object Identifier 10.1109/ICSMC.2000.886412

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)