

Ronaldo Ribeiro de Campos

Características de Sistemas Integrados de Gestão Empresarial desenvolvidos sob o modelo de *software* livre: informações para suporte à fase de seleção e viabilidade de instalação em pequenas empresas.

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Edson Walmir Cazarini

São Carlos
2006

*À minha esposa Elizabeth.
Aos meus filhos, Miguel e Uriel.
Aos meus pais, Moacir (em memória) e Maria.*

AGRADECIMENTOS

Ao Prof. Cazarini, pela dedicação, apoio e incentivo.

Aos colegas do grupo DECIDIR que colaboraram com sua experiência, conversas, incentivo e sugestões.

Ao pessoal do laboratório de Simulação e Jogos, pelo convívio, pela partilha do espaço e pela amizades conquistadas, em especial ao Jefferson pelo apoio na instalação dos programas.

Aos colegas funcionários do Departamento, Luiz Fernando, José Luis, Allan, Silvana, Sueli e Ruth.

Ao CNPQ, pelo apoio financeiro em parte desta pesquisa.

A Universidade de São Paulo pela infraestrutura que propiciou condições para a realização dessa pesquisa.

A todos aqueles que contribuíram de uma forma ou de outra para a realização deste trabalho.

A Deus, fonte de toda sabedoria.

RESUMO

CAMPOS, R.R. (2006). Características de Sistemas Integrados de Gestão Empresarial desenvolvidos sob o modelo de *Software* Livre : informações para suporte à fase de seleção e viabilidade de instalação em pequenas empresas. – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

Este trabalho realizou uma investigação descritiva nos sistemas integrados de gestão empresarial desenvolvidos sob o modelo de software livre. O objetivo foi obter um levantamento de informações que pudessem auxiliar na fase de seleção de um sistema de informação. O trabalho apresenta conceitos de *software* livre, sistemas de informação e informações sobre tecnologia em pequenas empresas sob os aspectos de equipamentos, pessoas e utilização de *software*. Foram abordados três aspectos na descrição dos sistemas: viabilidade de instalação, funcionalidades e usabilidade. O estudo foi realizado dentro de um referencial teórico qualitativo-positivista, com objetivo metodológico descritivo, utilizando como delineamento o levantamento. A base de dados para extração das amostras foi o *site SourceForge*, um repositório de *software* livre. A seleção de amostras foi feita em fases múltiplas e possibilitou uma análise sobre as características dos aplicativos de forma geral e depois uma análise mais detalhada dos programas selecionados. Os resultados apontam para a escassez de pessoal capacitado tanto para uso da tecnologia na pequena empresa quanto com experiência no desenvolvimento de *software* livre, mas revelam a viabilidade de instalação do *software* livre na pequena empresa, uma vez que o *hardware* necessário ao funcionamento dos sistemas está presente em seu ambiente, bem como a existência de sistemas integrados de gestão bastante adaptáveis às pequenas empresas.

Palavras-chave: sistemas integrados de gestão empresarial, ERP, pequenas empresas, *software* livre, sistema de informação.

ABSTRACT

CAMPOS, R.R. (2006). *Characteristics of Management Information Systems developed based on Free Software model: information to support in the phase of selection and viability of installation in small businesses.* – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

This present dissertation deals a descriptive investigation on integrated management information system based on free software model. The objective was to obtain a survey of information that could help the selection phase of an information system. The work presents concepts of free software, information systems and data on technology in small companies having in view equipment, peopleware and software use. Three aspects of the description of systems were analysed: viability of installation, functionalities and usability. The study was performed within a qualitative-positive theoretical reference, with a descriptive methodology objective and using the survey as delineation. The database for extraction of the samples was the SourceForge site, a repository of free software. The selection of samples was made in multiple phases and allowed for a general analysis of the characteristics of softwares and a detailed analysis of selected software. The results show a scarcity of people able for use technology in the small companies and experienced people in the development of free software. However the same results show viability of installing free software in the small company, as the necessary hardware is present in their environment, as well as the existence of integrated management systems sufficiently adaptable to the small companies.

Keywords: integrated management information system, ERP, small business, free *software*, information systems.

LISTA DE FIGURAS

Figura 1 – Ambiente da pesquisa	23
Figura 2 – Níveis de um sistema.....	29
Figura 3 - Componentes de um sistema e seus relacionamentos	30
Figura 4 – Sistema de informação	32
Figura 5 – Relacionamento: informação – decisão – ação	33
Figura 6 – Enfoque dos sistemas de informação ao longo do tempo	34
Figura 7 – Evolução dos sistemas ERP	39
Figura 8 – Arquitetura cliente-servidor em duas camadas.....	41
Figura 9 - Arquitetura cliente-servidor em três camadas	41
Figura 10 – Aplicações ERP acessam base de dados integrada.....	42
Figura 11 – Forças que dirigem a competição	50
Figura 12 – Integração: organização, sistema de informação e estratégia de negócios	53
Figura 13 - Categorias de <i>software</i> segundo a FSF	67
Figura 14 – Etapas de desenvolvimento de software	78
Figura 15 – Diagrama de um projeto de <i>software</i> livre	79
Figura 16 - Resultados do mecanismo de busca.....	126
Figura 17 - Atividades da seleção de amostras	128
Figura 18 – Resultados das atividades da seleção de amostras	133

LISTA DE GRÁFICOS

Gráfico 1 - Número de projetos no site SourceForge.....	22
Gráfico 2 - Modelo de computador utilizado nas MPE	99
Gráfico 3 - Sistemas operacionais utilizados nas MPE.....	102
Gráfico 4 - Outras categorias de <i>software</i> encontradas nas PE	106
Gráfico 5 - Empresas com área de própria de TI.....	108
Gráfico 6 - Número de pessoas na área de TI.....	108
Gráfico 7 - Estágio de Desenvolvimento dos Projeto.....	135
Gráfico 8 - Setores de aplicação.....	136
Gráfico 9 - Linguagens de programação utilizadas.....	137
Gráfico 10 - Sistemas operacionais utilizados pelos aplicativos	138
Gráfico 11 - Gerenciadores de Banco de Dados	139
Gráfico 12 - Idiomas dos aplicativos	141

LISTA DE QUADROS

Quadro 1 – Fases de implantação de um ERP	21
Quadro 2 – Metas, objetivos, funções e atividades do MRP.....	36
Quadro 3 - Características dos sistemas ERP	45
Quadro 4 – Liberdades garantidas pelas licenças de <i>Software</i> Livre X Open <i>Source</i>	66
Quadro 5 - Principais característica de direito autoral e copyright	72
Quadro 6 – Principais licenças de <i>software</i> livre citadas	75
Quadro 7 - Exemplos de projetos de <i>Software</i> Livre.....	81
Quadro 8 - Vantagens e desvantagens do <i>software</i> livre	88
Quadro 9 - Média de computadores por porte de empresa em 2003	99
Quadro 10 – Aplicativos selecionados	143
Quadro 11 - Aplicativos selecionados segundo a linguagem de programação	144
Quadro 12 - Informações Compiere.....	150
Quadro 13 – Informações Evaristo	151
Quadro 14 - Informações Ezybiz.....	153
Quadro 15 – Informações FacturaLux	155
Quadro 16 - Informações Freedom.....	157
Quadro 17 - Informações Oratio	160
Quadro 18 - Informações Sequóia ERP.....	162
Quadro 19 - Informações Tiny ERP	164
Quadro 20 – Informações webERP	167
Quadro 21 – Informações University.....	168
Quadro 22 – Informações Value	169
Quadro 23 – Nível de dificuldade de instalação.....	171
Quadro 24 – Software adicional para cada aplicativo instalado.....	172

Quadro 25 – Presença de programa de instalação / desinstalação.....	173
Quadro 26 – Acesso a banco de dados.....	174
Quadro 27 – Adequação de dados	175
Quadro 28 - Módulos	176
Quadro 29 – Tipos de relatórios.....	177
Quadro 30 – Características dos relatórios.....	178
Quadro 31 – Segurança de acesso	179
Quadro 32 – Aspectos de usabilidade	180
Quadro 33 – Idiomas e formas de suporte.....	181
Quadro 34 - Características gerais dos aplicativos.....	182
Quadro 35 – Hardware requerido e o ambiente da pequena empresa	184
Quadro 36 – Adequação módulos às áreas da pequena empresa.....	184
Quadro 37 – Pontos favoráveis e desfavoráveis à instalação de <i>software</i> livre nas PE	187

LISTA DE TABELAS

Tabela 1 – Classificação europeia de micro, pequena e média empresa.....	94
Tabela 2 – Parâmetros de Pequenas empresas segundo a SBA.....	95
Tabela 3 – Classificação de empresas quanto ao número de empregados	95
Tabela 4 – Classificação de empresas quanto ao faturamento	96
Tabela 5 - Sistemas Operacionais utilizados em servidores.....	101
Tabela 6 - Funções utilizadas nos servidores nas pequenas empresas.....	101
Tabela 7 - Gerenciadores de Banco de dados utilizados nas pequenas empresas.....	102
Tabela 8 – Forma de acesso à internet nas pequenas empresas.	103
Tabela 9 - Classificação de projetos segundo <i>SourceForge</i>	134
Tabela 10 - Classificação das áreas de aplicação do <i>software</i>	135
Tabela 11 – Linguagens de programação utilizadas.....	137
Tabela 12 - Relação Percentual de utilização de Sistema Operacional.....	139
Tabela 13 - Ferramentas de acesso ao banco de dados.....	140
Tabela 14 - Número de ocorrências de idiomas	142

LISTA DE SIGLAS

AMP	Apache – Mysql / Perl / Python
ASP	<i>Application Service Provider</i>
BI	<i>Business Intelligence</i>
BSD	<i>Berkeley Software Distribution</i>
CAD	<i>Computer Aided Design</i>
CMS	<i>Content Management System</i>
CRM	<i>Customer Relationship Management</i>
CRP	<i>Customer-centric Resource Planning</i>
CSV	<i>Comma Separated Value</i>
CVS	<i>Concurrent Version System</i>
EIS	<i>Executive Information System</i>
ERM	<i>Enterprise Resources Management</i>
ERP	<i>Enterprise Resource Planning</i>
ES	<i>Enterprise System</i>
FDL	<i>Free Document License</i>
FSF	<i>Free Software Foundation</i>
FSFE	<i>Free Software Foundation of Europe</i>
GCC	<i>GNU Compiler License</i>
GNU	<i>GNU's is not Unix</i>
GPL	<i>General Public License</i>
GUI	<i>Grafic User Interface</i>
HTML	<i>Hipertext Markup Language</i>
ICMS	Imposto sobre Circulação de Mercadorias e Serviços
IPI	Imposto sobre Produto Industrializado
JDBC	<i>Java Database Connectivity</i>
JDK	<i>Java Development Kit</i>

JSP	<i>Java Server Pages</i>
LAMP	<i>Linux – Apache – Mysql / Perl / Python</i>
LGPL	<i>Less General Public License</i>
MAMP	<i>Machintosh – Apache – Mysql / Perl / Python</i>
MB	<i>Megabyte</i>
MHZ	<i>Megahertz</i>
MIS	<i>Management Information System</i>
MIT	<i>Massachusetts Institute of Technology</i>
MPL	<i>Mozilla Public License</i>
MRP	<i>Material Requirement Planning</i>
MRP II	<i>Manufacturing Resoruce Planning</i>
NPL	<i>Netscape Public License</i>
OSI	<i>Open Source Initiative</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Personal Home Page</i>
POSIX	<i>Portable Operating System</i>
RAM	<i>Randomic Access Memory</i>
SCM	<i>Supply Chain Management</i>
SGBD	Sistema Gerenciador de Banco de Dados
SGE	Sistema de Gestão Empresarial
SI	Sistema de Informação
SIG	Sistema Integrado de Gestão ou Sistema de Informação Gerencial
SIGE	Sistema Integrado de Gestão Empresarial
SQL	<i>Structured Query Language</i>
TI	Tecnologia da Informação
USL	<i>Unix System Laboratories</i>
XML	<i>Extensible Markup Language</i>
XRP	<i>EXtended Resource Planning</i>
WAMP	Windows – Apache – Mysql / Perl / Python
WFM	<i>Work-Flow Management</i>

SUMÁRIO

1 - Introdução	17
1.1 Caracterização do tema da pesquisa	17
1.2 O problema de pesquisa	20
1.3 Justificativa	24
1.4 Objetivo	25
1.5 Apresentação do trabalho	26
Capítulo 2 - Sistemas Integrados de Gestão Empresarial	27
2.1 Sistema	27
2.2 Sistema de informação e sistema de informação gerencial	31
2.3 Histórico dos sistemas de gestão empresarial	33
2.3.1 MRP - Material Requirement Planning	34
2.3.2 MRP II - <i>Manufacturing Resource Planning</i>	37
2.3.3 ERP e Sistemas Integrados de Gestão Empresarial	39
2.4 ERP e a era da Internet	46
2.5 Abrangência da integração	47
2.6 Estratégia, vantagem competitiva e tecnologia de informação	48
Capítulo 3. Software Livre e Software de Código aberto	55
3.1 <i>Software</i> Livre	55
3.1.1 Histórico: Projeto GNU, <i>Free Software Foundation</i> e Linux	56
3.1.2 Definição de <i>Software</i> Livre	59
3.2 <i>Software</i> de Código Aberto	61
3.2.1 Histórico da <i>Open Source Initiative</i>	62
3.2.2 Definição de Código Aberto e seus termos de distribuição	63
3.3 Diferença entre <i>Software</i> Livre e <i>Open Source</i>	65
3.4 Categorias de <i>Software</i> segundo a FSF	67
3.5 Direito autoral e <i>Copyright</i>	69

3.6 Regras quanto à distribuição e tipos de licença	72
3.7 Projetos de <i>Software</i> Livre	78
3.8 Utilização: Vantagens e desvantagens	83
3.8.1 Vantagens.....	85
3.8.2 Desvantagens.....	87
3.8.3 Desvantagens imaginárias.....	88
3.9 <i>Software</i> livre e governo.....	90
3.10 Aspectos sociais.....	91
Capítulo 4 - Pequenas empresas.....	93
4.1 Características das pequenas empresas	93
4.2 Pequenas empresas e tecnologia de Informação	97
4.3 Pequenas empresas e <i>softwares</i> de gestão.....	104
4.4 Pequenas empresas e recursos humanos na área de informática.....	106
4.5 Considerações	109
Capítulo 5 – Procedimentos metodológicos	113
5.1 Metodologia.....	113
5.1.2 Método.....	113
5.2 As variáveis de pesquisa.....	114
5.2.1 Requisitos de instalação.....	115
5.2.2 Funcionalidades.....	116
5.2.3 Usabilidade	118
5.3 O Questionário	120
5.3.1 Parte 1	120
5.3.2 Parte 2	122
5.3.3 Parte 3	123
5.4 Seleção de amostras.....	124
5.5 Instalação de aplicativos e coleta de dados	128
5.6 Relacionamento com dados das pequenas empresas.....	129
5.7 Limitações	129
Capítulo 6 – Resultados	131
6.1 Fases da seleção de amostras.....	131
6.2 Panorama geral das características dos aplicativos.....	134
6.2.1 Estágio de desenvolvimento dos projetos.....	134
6.2.2 Setor de aplicação	135

6.2.3	Linguagens de programação	136
6.2.4	Sistema operacional	138
6.2.5	Banco de dados	139
6.2.6	Idiomas	140
6.3	<i>Softwares</i> selecionados.....	142
6.3.1	Pré-requisitos para instalação	144
6.3.2	Compiere	146
6.3.3	Evaristo.....	150
6.3.4	EzyBiz.....	152
6.3.5	FacturaLux Lite	154
6.3.6	Freedom	156
6.3.7	Linux Kontor.....	157
6.3.8	MicroERP.....	158
6.3.9	openXpertya	158
6.3.10	Oratio	159
6.3.11	Sequóia ERP	161
6.3.12	Tiny ERP.....	162
6.3.13	WebERP	164
6.3.14	University	168
6.3.15	Value.....	169
6.4	Viabilidade de instalação.....	170
6.4.1	Nível de dificuldade de instalação	170
6.4.2	Necessidade de software adicional	171
6.4.3	Recursos de instalação.....	172
6.5	Funcionalidades	173
6.5.1	Acesso a banco de dados.....	173
6.5.2	Adequação de dados à empresa.....	174
6.5.3	Módulos	175
6.5.4	Relatórios e gráficos	176
6.5.5	Integração com outros softwares.....	178
6.5.6	Segurança	179
6.6	Usabilidade.....	180
6.7	Viabilidade de instalação nas pequenas empresas.....	183
6.7.1	Equipamento requerido.....	183

6.7.2 Áreas funcionais	184
6.7.3 Pessoas	185
7 - Considerações finais.....	188
7.1 Conclusão	188
7.2 Sugestões para estudos futuros.....	191
BIBLIOGRAFIA	194
APÊNDICE A - MODELO DE QUESTIONÁRIO	208
APÊNDICE B – RELAÇÃO DE <i>SOFTWARES</i>	215
ANEXO A – RELAÇÃO DE LICENÇAS DE <i>SOFTWARE LIVRE</i>.....	224

1 - Introdução

Este capítulo apresenta a caracterização do tema da pesquisa, seu objetivo e a forma como esse texto está estruturado.

1.1 Caracterização do tema da pesquisa

A idéia do desenvolvimento de *software* livre surgiu na década de 80 e seu objetivo inicial era desenvolver um sistema operacional que fosse totalmente portátil e livre. Livre sob o aspecto do programa poder ser distribuído com seu código fonte, para que pudesse ser adaptado às características de cada organização em particular. Livre sob um conceito filosófico de liberdade, mas que não significa necessariamente que por isso não se deva pagar para obtê-lo.

Com o passar do tempo a idéia se difundiu e teve em particular uma derivação, com algumas variações em relação à forma de licença dos programas, denominada de *Open Source* (Código Aberto).

O objetivo inicial de desenvolver um sistema operacional foi superado e existe um número crescente de projetos de *software* dedicados a várias outras áreas: sistemas gerenciadores de banco de dados, servidores de página *web*, linguagens de programação de alto nível, gerenciamento de projetos e gestão empresarial (*SOURCEFORGE*, 2005; *BEHLENDORF*, 1999).

As vantagens dos programas de código aberto, dentre elas o baixo custo de aquisição, são um atrativo às pequenas empresas que possuem recursos tecnológicos, mas pouco recurso financeiro e buscam uma forma de utilizar a tecnologia de informação como forma de obter vantagens competitivas (*CAMPOS e CAZARINI*, 2005a).

O Sistema Integrado de Gestão Empresarial é apontado como uma das opções de uso da tecnologia de informação, como auxílio à obtenção de

vantagem competitiva e existem projetos de desenvolvimento deste tipo de programa sob os moldes do *software* livre.

O que este trabalho trata como Sistemas Integrados de Gestão se confunde com a sigla ERP (*Enterprise Resource Planning*), que passa pelo conceito de seus antecessores, o MRP (*Material Requirement Planning*) e o MRP II (*Manufacturing Resources Planning*).

Os sistemas ERP são a evolução dos sistemas MRP II, que por sua vez evoluíram a partir do MRP.

O MRP foi um sistema criado com o objetivo de controlar o estoque e dar apoio às funções de planejamento de produção e compras (COLANGELO FILHO, 2001). Ainda de acordo com Colangelo Filho (2001), seus procedimentos eram baseados em listas de materiais e um plano de produção. O MRP II expandiu o MRP e englobou aspectos financeiros, orçamentos e custeio de produção, mas apesar desta expansão ainda não havia uma integração completa com os outros sistemas da empresa e o suporte completo e integrado ao processo de negócios ainda era difícil.

Na década de 90, o MRP II sofreu novas modificações e sua área de aplicação tornou-se ainda mais abrangente. Sua evolução, o ERP, é capaz de trabalhar em um ambiente cliente-servidor, com uma base de dados única e integrada, o que possibilita dar à empresa uma visão mais ampla da utilização dos recursos em diversas áreas.

Mendes (2003) descreve os sistemas ERP como tendo grande abrangência funcional, procurando incorporar todas as atividades possíveis realizadas por uma organização. A estrutura do sistema ERP é normalmente dividida em módulos que podem ser instalados separadamente. Os módulos mais comumente encontrados em sistemas disponíveis no mercado são os seguintes:

- Módulos relacionados a operações e gerenciamento da cadeia de suprimentos;
- Módulos relacionados à gestão financeira, contábil e fiscal;
- Módulos relacionados à gestão de recursos humanos;
- Módulos relacionados ao MRP II (planejamento de materiais, capacidade, gestão de demanda, compras e vendas)

Colangelo Filho (2001) registra que, à medida que o ERP foi ampliando suas áreas de atuação dentro das organizações, acabou por determinar a classificação para uma determinada categoria de *software*, de forma que é possível encontrar referências a programas ditos “ERP para comércio”, ou “ERP para serviços”.

Davenport (2002) explica que, a partir desta visão, a sigla ERP deixou de ser a denominação mais adequada, refletindo apenas a origem destes sistemas na área de produção, uma vez que se trata de uma alteração da sigla MRP. Para Davenport (2002), esses sistemas são denominados de “sistemas de gestão empresarial (SGE), ou de ES, da sigla em inglês para *enterprise systems*, ou ainda Sistemas Integrados de Gestão (SIG), ou mesmo utilizando a sigla ERP.

Assim, chega-se a denominação de Sistemas de Gestão Empresarial como uma referência para o que se chamava de ERP e, em muitos casos, as expressões ainda são usadas como sinônimos, acrescentado que, para Padilha e Marins (2005), no Brasil são também conhecidos como Sistemas Integrados de Gestão Empresarial (SIGE).

A implantação de um sistema de informação deste tipo é um projeto complexo, que atinge a organização como um todo, envolvendo questões de processos, tecnologia, pessoas e as conseqüentes mudanças organizacionais ocorridas (MENDES, 2003).

Durante esse complexo processo de implantação, existem diferentes fases pelas quais a empresa deve passar. Uma vez tomada a decisão de implantação, o primeiro passo para uma implantação com sucesso é selecionar o sistema a ser implantado (CORRÊA, 1998).

A fase de seleção é o período em que se realiza uma análise de adequação de funcionalidades do sistema, na busca de identificar se as características particulares de cada empresa são atendidas pelo sistema.

A dificuldade é identificar dentre o crescente número de projetos de software livre, quais são os SIGE que se destinam à utilização em determinadas áreas de atividade econômica, quais são os requisitos necessários à sua instalação e quais são as funcionalidades presentes nesses sistemas.

Esse trabalho considera a situação das pequenas empresas na busca de alcançar vantagens competitivas com a utilização da tecnologia, cogitando a utilização de SIGE no modelo do *software* livre, e apresenta uma investigação descritiva de tais aplicativos com o objetivo de reunir informações que construam uma visão dos requisitos de instalação e funcionalidades presentes nestes programas podendo ser útil na fase de seleção do sistema a ser escolhido.

1.2 O problema de pesquisa

A abrangência dos aplicativos desenvolvidos sob o modelo de *software* livre atingiu diferentes áreas: sistemas gerenciadores de banco de dados, servidores de página *web*, linguagens de programação de alto nível, gerenciamento de projetos e gestão empresarial, e não mais se limita apenas ao sistema operacional. (*SOURCE FORGE*, 2005).

O *software* livre apresenta vantagens em relação ao custo, robustez, independência de tecnologia ou fornecedor único e possibilidade de personalização.

Estas vantagens, em especial o custo, atraem a atenção das pequenas empresas que são possuidoras de recursos tecnológicos, pois conforme descrito por SEBRAE (2003), 47 % das micro empresas possuem computador, índice este que se eleva para 61% quando se trata de indústrias, e chega a 81% quando se trata de empresas de pequeno porte.

O número de computadores justifica-se pela tentativa de utilização dos recursos de tecnologia de informação como vantagem competitiva. De acordo com Zimmerer, Scarborough (1994), aquelas pequenas empresas que aprendem a explorar o uso da informação e tecnologia para atender melhor seus clientes, conseguem obter ganhos de competitividade, assim como as pequenas empresas que coletam e fazem uso da informação disponível para gerenciar suas operações de forma mais eficiente, também conseguem uma vantagem competitiva. A tecnologia está disponível para impulsionar a produtividade, controlar estoques e fluxo de caixa, identificar necessidades e preferências de clientes e antecipar problemas e oportunidades.

É nesse ponto que os SIGE passam a ser vistos como uma solução que pode auxiliar na busca de “vantagens competitivas”, atuando principalmente sobre a redução de custos à medida em que auxilia tarefas de planejamento de compras, controle de estoque, cadastro de clientes e produtos.

A adoção de sistemas ERP não é um processo simples, que atinge apenas uma área da organização, ele envolve questões de processos, tecnologia, pessoas e as conseqüentes mudanças organizacionais ocorridas (MENDES, 2003).

Para Corrêa (1998), o sucesso na adoção de um ERP se inicia na sua seleção, período em que se realiza uma análise de adequação de funcionalidades com o objetivo de verificar se as particularidades de cada organização são atendidas pelo sistema. A tarefa de seleção representa apenas uma das atividades da fase de pré-implantação do sistema. Cerri (2004) apresenta informações que demonstram as fases de implantação de um sistema ERP. O quadro 1 apresenta uma visualização de qual parte destas fases de instalação este trabalho se concentra.

Fases	Atividades
Pós-implantação	Estabilização e materialização dos benefícios Aplicações integradas a e-bussines Atualizações do sistema
Implantação	Planejamento da implantação Desenho da solução Testes e implantação
Pré-implantação	Estudo de viabilidade Seleção de parceiros Seleção de produtos 

Quadro 1 – Fases de implantação de um ERP

Fonte: Adaptado de Cerri, (2004, p.82)

É na fase de seleção que se iniciam as dificuldades com relação à seleção de um SIGE nos modelos de *software* livre, principalmente por causa do crescente número de projetos deste tipo. Uma pesquisa na data de 04/01/2005, em *sites* onde podem ser cadastrados os projetos de *software* livre, mostra o seguinte:

- O site “*Sourceforge.net*”¹, o maior repositório de projetos do mundo, *Sourceforge* (2005), registrava a quantia de 58.484 projetos, dos quais 3.672 ligados à categoria Escritório/Negócios, e dentre esses, 1.169 pertencentes à subcategoria Finanças, que é o grupo onde se encontram também a categoria ERP ou sistemas integrados de gestão.
- O site “*Freshmeat*”², outra fonte de pesquisa sobre *softwares* *Freshmeat* (2005), possuía 26.623 projetos, sendo 1.861 cadastrados na categoria Escritório/Negócios, com 561 pertencentes à subcategoria Finanças.

A mesma pesquisa, realizada em Maio de 2005, registrou 110.650 projetos cadastrados no site *Sourceforge*, sendo 4.272 pertencentes à categoria Escritório/Negócios e 1.266 na subcategoria Finanças. Embora um projeto possa estar cadastrado em mais de uma categoria, o crescimento do número de projetos é bastante considerável. O gráfico 1 ilustra o crescimento do número de projetos.

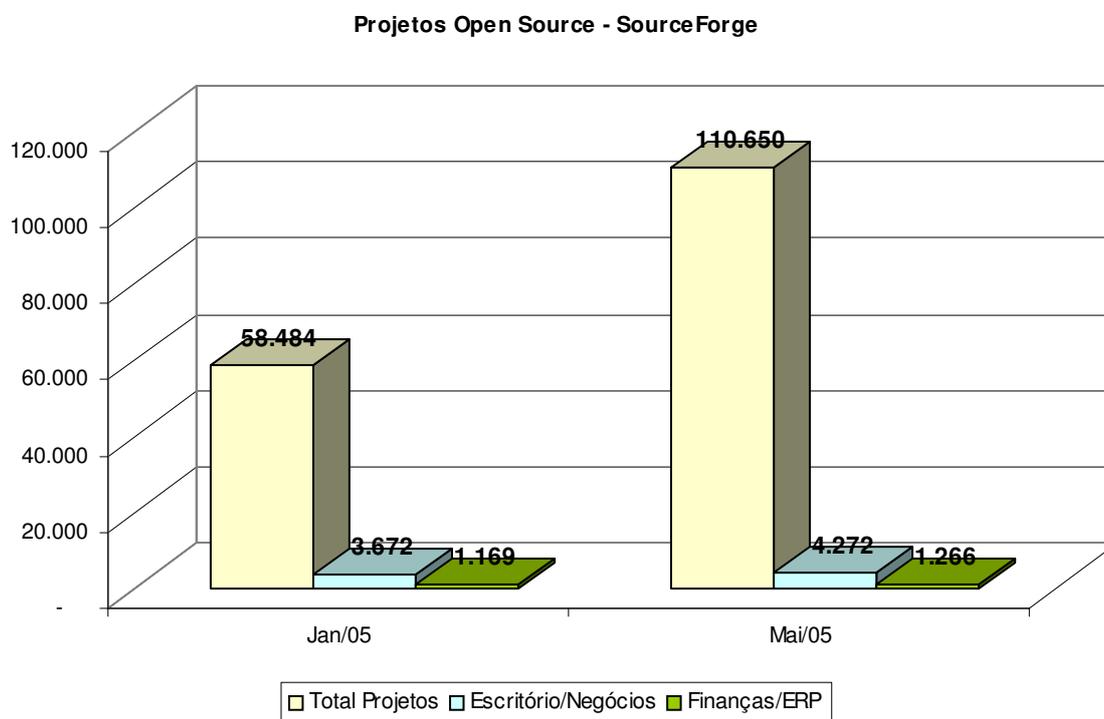


Gráfico 1 - Número de projetos no site SourceForge

¹ disponível em: www.sourceforge.net

² disponível em: www.freshmeat.net

Considerando o ambiente de tecnologia das pequenas empresas como propício à instalação de um sistema integrado de gestão empresarial e de outro lado a oferta existente pelo crescente número de projetos de software livre, diante da necessidade de um processo de implantação que passa necessariamente pela fase de seleção do sistema, tem-se o cenário (ilustrado pela Figura 1) em que se baseia esta pesquisa

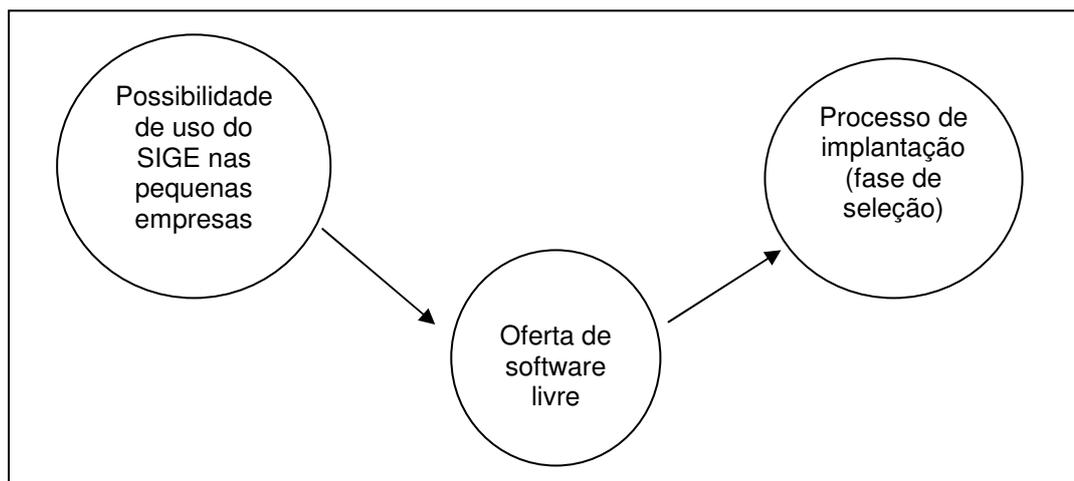


Figura 1 – Ambiente da pesquisa

Assim, a definição do problema de pesquisa envolve, de um lado, as pequenas empresas na busca da implantação de SIGE e, de outro lado, a existência de um número crescente de *software* de código aberto em estágios diferentes de desenvolvimento e com características diferentes. A partir dessa situação foram estabelecidas as questões de pesquisa diante da dificuldade de identificar:

- **quais são as características dos projetos de sistemas integrados de gestão empresarial desenvolvidos sob os moldes do software livre quanto a requisitos de hardware, principais funcionalidades e usabilidade?**
- **o ambiente de tecnologia de informação das pequenas empresas está apto à instalação destes sistemas?**

1.3 Justificativa

Parte da justificativa se encontra descrita no item anterior, quando foi exposto o problema de pesquisa, mas, além disso, outros fatores podem ser apresentados:

- Redução do custo de aquisição de *software* gerado pelo uso de *software* livre que dispensa o pagamento de licenças. De acordo com Zimmerer, Scarborough (1994) um dos principais fatores que as pequenas empresas enfrentam durante a informatização são os custos gerados pela aquisição de *software*;
- O crescente número de projetos de *software* de código livre dedicado à gestão de empresas;
- O trabalho de identificar os aplicativos que estão qualificados como SIGE, descobrir quais estão em um estágio de desenvolvimento maduro o bastante para serem usados e ainda identificar quais são os requisitos mínimos para que eles possam ser colocados em operação, bem como as suas principais funcionalidades presentes, é uma tarefa que economizaria tempo e dinheiro para a pequena empresa que está à procura de um *software* desse tipo;
- Fornecer informações à comunidade de *software* livre sobre os diversos projetos de SIGE desenvolvidos;
- Contribuir para os trabalhos de pesquisa relacionados ao *software* livre, pois segundo Augusto (2003, p. 2) existem poucas pesquisas científicas sobre o assunto.
- Indicações de que a tendência é o que ERP (hoje ainda pouco utilizado na pequena empresa) passe a ter um maior uso na média empresa e seja adotado também na pequena empresa.
- Contribuir para o trabalho dos colegas que integram o grupo destinado à linha de pesquisa: Gestão estratégica da tecnologia de informação, em particular para o tema: gestão de tecnologia de informação na pequena empresa.

Outro aspecto relevante é a diferenciação deste trabalho em relação à outros já realizados, uma vez que embora já tenham sido feitas pesquisas como a de Reis (2003) abordando o processo de desenvolvimento de Software Livre; a de Augusto (2003) que trata das motivações de usuários e desenvolvedores do *Software Livre*; de Saleh (2004) sobre adoção de tecnologia e a de Krishnamurthy (2002) que tratou do perfil de usuários envolvidos em projetos, entre outras, não foi encontrado nenhum trabalho que tratasse de forma específica os projetos de sistemas integrados de gestão desenvolvidos na comunidade de *software livre*.

Além destes aspectos, existe ainda um interesse particular por parte do autor tanto no assunto pequenas empresas quanto software livre. O autor teve a oportunidade de atuar como programador em uma pequena empresa durante um período de dez anos e acompanhar o crescimento da estrutura de tecnologia de informação que foi apoiada por um aumento significativo do número de equipamentos de quatro para quarenta, pela chegada de uma estrutura de rede de computadores, da *internet* e da constante busca por um sistema integrado de gestão que suprisse as necessidades da empresa. Embora o desenvolvimento do sistema de gestão fosse feito pela própria empresa, o servidor de arquivos e o servidor de banco de dados utilizados eram *software livre*.

1.4 Objetivo

O objetivo principal desse trabalho é:

apresentar um levantamento das principais funcionalidades dos sistemas integrados de gestão empresarial desenvolvidos sob os moldes do *software livre*, avaliar o grau de maturidade para uso e analisar a viabilidade de instalação na pequena empresa.

1.5 Apresentação do trabalho

Esse trabalho encontra-se dividido em sete capítulos, sendo que a Introdução constitui o capítulo um e apresenta a caracterização do tema da pesquisa, justificativa e objetivo, bem como a forma como o trabalho está estruturado.

Os capítulos dois, três e quatro apresentam uma revisão bibliográfica sobre os três principais assuntos referentes à pesquisa: sistema de informação, *software* livre e pequenas empresas.

Os capítulos cinco, seis e sete tratam da metodologia utilizada, dos resultados e da conclusão. Os capítulos são explicados em mais detalhes a seguir:

O capítulo dois apresenta as fases de evolução pelas quais passou o sistema de informação empresarial para que chegasse em seu estágio atual, desde a definição do que é um sistema, passando pelas fases do MRP, MRP II, MRP III e ERP. São tratadas também as questões de como utilizar um sistema de informação para alcançar vantagem competitiva e a necessidade da integração do sistema à empresa.

O capítulo três apresenta informações e conceitos sobre *software* livre e *software* de código aberto: quais as diferenças, quais os tipos de licenças, e quais os critérios de classificação. São apresentadas as vantagens e desvantagens do *software* livre e uma breve discussão sobre a questão dos direitos autorais e do *copyright*.

No capítulo quatro, são tratadas as definições de pequenas empresas adotadas na Europa, nos Estados Unidos e no Brasil, bem como sua estrutura de tecnologia de informação com base na visão sócio-técnica, onde o sistema de informação é um conjunto de tecnologia, sistemas e pessoas.

As questões metodológicas adotadas para este trabalho, variáveis de pesquisa e limitações são apresentadas no capítulo cinco.

O capítulo seis apresenta os resultados obtidos dividindo-se em resultados gerais, características dos aplicativos que foram instalados e uma consideração teórica sobre a possibilidade de instalação dos aplicativos selecionados no ambiente da pequena empresa. A conclusão e sugestões para trabalhos futuros são apresentadas no capítulo sete.

Capítulo 2 - Sistemas Integrados de Gestão Empresarial

Os Sistemas Integrados de Gestão Empresarial são o resultado de uma evolução dos sistemas de informação aplicados às organizações. Este capítulo apresenta o histórico dessa evolução, desde o conceito de sistema. Também é apresentada a necessidade de integração do sistema aos objetivos da empresa para que o sistema possa realmente ser utilizado como uma ferramenta para alcance da vantagem competitiva.

2.1 Sistema

De acordo com Mañas (1999), foi entre os anos de 1820 e 1904 que viveu o filósofo inglês chamado Herbert Spenser. Este filósofo sustentava a teoria de que um organismo social é semelhante a um organismo individual, pois apresentam traços de crescimento e podem se tornar mais complexos de forma que suas partes apresentem uma interdependência mútua, além do fato de existir crescente interação e heterogeneidade entre as partes que compunham o organismo.

Mañas (1999) registra que no início da década de 30, Claude Lévi-Strauss, filósofo e antropólogo belga, defendia que uma determinada estrutura se caracterizava como “sistema” quando os elementos que a compunham estavam combinados de tal forma que uma modificação realizada em um dos elementos determinaria modificações em todos os outros.

O termo “sistema” também foi utilizado pelo biólogo alemão Ludwing von Bertalanffy, na década de 40, para definir organismos vivos por ele estudados. Von Bertalanffy verificou, através de seus estudos, que apesar de muitas das características dos organismos serem diferentes, em quaisquer um dos tipos existiam características se mantinham comuns. O próximo passo

de Von Bertalanffy foi estender suas observações à outros tipos de organismos, entre eles os organismos mecânicos e os sociais. A constatação foi que também nestes organismos algumas características se mantinham comuns, independentemente da natureza do sistema CAUTELA E POLLONI (1996b), MAÑAS (1999) e BIO (1996).

Com base na observação de que o organismo era composto por uma série de elementos diferentes que interagem entre si com a finalidade de atingir o objetivo, o biólogo alemão formulou o que passou a ser conhecido como Teoria Geral dos Sistemas, considerando que a característica mais importante em todos os organismos era a identidade, ou seja, o objetivo que o organismo buscava alcançar (CAUTELA e POLLONI, 1996).

O conceito definido por Von Bertalanffy, segundo informa Mañas (1999) é o de sistema aberto, pois em um sistema fechado nenhuma influência entra ou deixa o sistema, enquanto que o sistema aberto importa e exporta influências que alteram seus componentes. Desta forma, Mañas (1999, p.62) define um sistema aberto como o “conjunto de partes em constante interação, constituindo um todo orientado para determinados fins e em permanente relação de interdependência com o ambiente externo”.

Ainda com referência à sistemas, seguem-se algumas outras definições:

- “um conjunto de partes integrantes e interdependentes que, conjuntamente, formam um todo unitário com determinado objetivo e efetuam determinada função” OLIVEIRA (2004, p. 23);
- “a disposição das partes de um todo, dentro de uma estrutura organizada, com a finalidade de executar tarefas” (CRUZ, 2003, p.55);
- “uma entidade composta de dois ou mais componentes ou subsistemas que interagem para atingir um objetivo comum” (GIL 1999, p.13)

Em se tratando de processamento de dados, segundo Gil (1999, p.13), “o termo sistema diz respeito a um conjunto de equipamentos ou conjunto de programas”.

Pode-se assim notar que as definições utilizadas levam em consideração as partes que compõe o sistema. Vale lembrar que estas diferentes partes podem ser outros sistemas dentro do próprio sistema,

recebendo o nome de subsistemas. Nota-se também pelo conceito de “sistema aberto” aplicado ao sistema de informação que este está inserido em um ambiente.

Para tornar mais claro o que significa, neste caso, o conceito de ambiente, toma-se por base a definição de Oliveira (2004), onde o ambiente de um sistema é um conjunto de elementos que não pertencem ao sistema, mas cujas alterações podem provocar mudanças ou alterações nos elementos que compõe o sistema, assim como alterações nos elementos que compõe o sistema podem provocar alterações no ambiente.

Na Figura 2, é possível visualizar o sistema sob três perspectivas ou níveis de hierarquia. Tomando como referência o centro da figura, pode-se identificar:

- Subsistemas, como as partes que integram o sistema (sistemas dentro do sistema);
- Sistema como o conjunto de elementos que está sendo considerado, ou estudado;
- Supersistema ou ambiente, onde o sistema passa a ser considerado como um subsistema do todo.

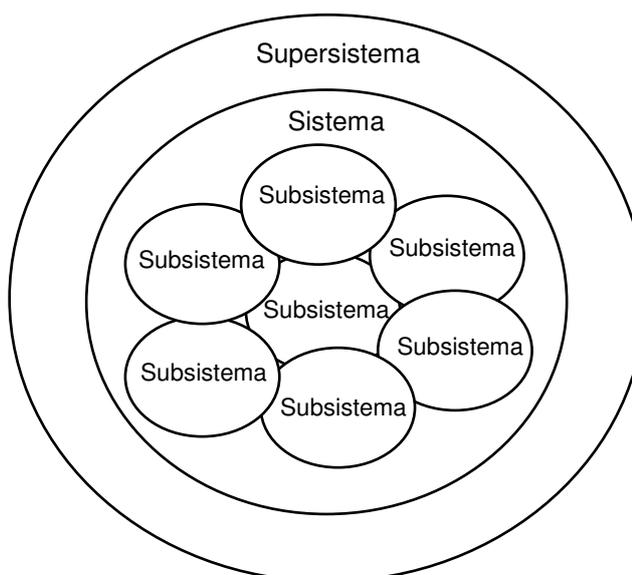


Figura 2 – Níveis de um sistema

Fonte: Adaptado de Oliveira (2004, p.26)

Outra característica dos sistemas é relatada por Oliveira (2004), ao informar que o sistema possui uma forma de funcionamento baseada em:

- **Objetivo:** é a própria razão de existência do sistema, é a finalidade para a qual ele foi criado.
- **Entradas:** é o material fornecido ao sistema para que este possa realizar suas operações, gerando assim uma saída.
- **Saídas:** são os resultados do processo de transformação realizada pelo sistema. As saídas podem ser vistas como a finalidade para a qual se reuniram os esforços, portanto devem estar de acordo com o objetivo.
- **Controles e avaliação:** ferramentas para verificar se as saídas estão de acordo com o objetivo. Isto pressupõe que as saídas devem ser quantificáveis, e a existência de um valor padrão para que o controle possa comparar os valores das saídas e determinar se estão ou não de acordo com o padrão.
- **Retroalimentação (feed-back):** processo que acontece a cada nova saída e que pode ser considerado como a reintrodução de uma saída sob a forma de uma entrada. A retroalimentação possibilita que o resultado das operações do sistema seja reincorporado a este, causando uma reação em função da nova informação.

A Figura 3 ilustra os elementos e características de um sistema, bem como seus relacionamentos.

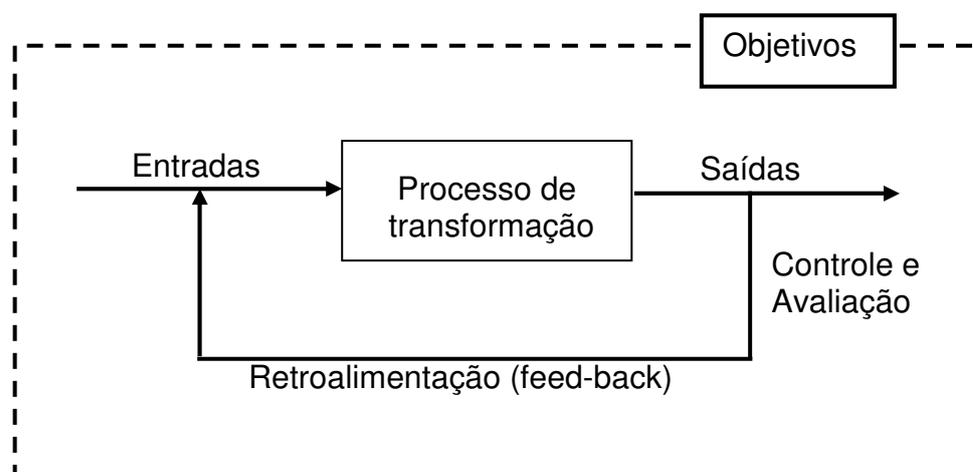


Figura 3 - Componentes de um sistema e seus relacionamentos

Fonte: Oliveira (2004, p.24)

A partir da definição formada sobre o conceito de sistema, esta passou a ser utilizada em outros setores além dos organismos vivos, atingindo o mundo empresarial e as questões relacionadas à informação, surgindo os conceitos de organização como um sistema, sistemas de informação e sistemas de informação gerencial.

2.2 Sistema de informação e sistema de informação gerencial

Nesta seção, serão apresentados conceitos sobre Sistema de Informação (SI) e Sistema de informação Gerencial (SIG).

Segundo Jobim Filho (1979, p. 9), sistema de informação é “um subsistema da organização interessado no registro, armazenamento, fluxo, tratamento, divulgação e outros usos que possam ser feitos da informação”.

Para Gil (1999, p. 14), o sistema de informação é composto por um conjunto de recursos humanos, materiais, tecnológicos e financeiros agrupados segundo uma seqüência lógica que permita o processamento de dados e sua correspondente tradução em informação. Outra definição apresentada por Gil (1999) é a de que um sistema de informação é constituído por pelo menos uma pessoa que possui características psicológicas e que enfrenta um problema dentro de um ambiente cuja resolução necessita de dados. A finalidade da obtenção destes dados é obter uma solução criada a partir das informações obtidas com o tratamento dos dados adquiridos e distribuídas sob um modo de apresentação.

Ao se aplicar esta definição ao modelo de componentes de sistema, tem-se o resultado apresentado na Figura 4, onde pode-se observar que por meio da atividade de processamento ocorre a transformação de dados em informação que será o resultado buscado pelo sistema de informação.

Para facilitar a compreensão da Figura 4, faz-se necessário uma definição do que é considerado dado e o que considera-se informação. A definição é dada pelo próprio Gil (1999, p. 13), ao explicar que o dado é a matéria-prima com que o sistema de informação vai trabalhar e informação é o produto final, ou seja, são os dados depois de processados e apresentados sob a forma, prazo e conteúdo adequado ao usuário do sistema de informação.

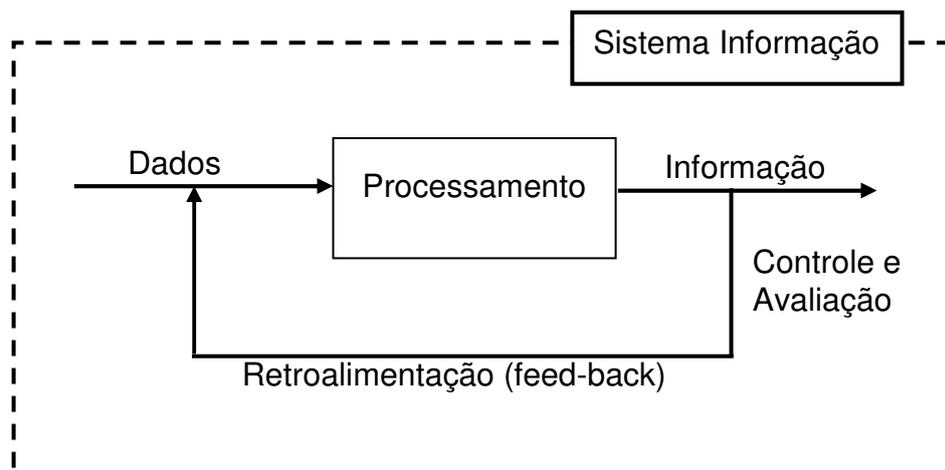


Figura 4 – Sistema de informação

Fonte: Adaptado de Gil (1999, p.13)

Cautella e Polloni (1996a) registram que o SIG (Sistema de Informação Gerencial) tradução da sigla MIS (Management Information System) nasceu da idéia de desenvolvimento de um sistema de informação que pudesse prover informações gerenciais integradas, oferecendo além de informações operacionais a possibilidade de informações destinadas ao nível executivo da organização. Para Cautella e Polloni (1996a), a definição de Sistema de Informação Gerencial pode ser apresentada como: “Um sistema de informação que englobe todos os componentes da organização e todos os seus níveis”, considerando que o SIG envolve o conjunto de informações da empresa, sejam elas automatizadas ou não.

Cautella e Polloni (1996a) também apresentam uma distinção entre sistema de informação, que seria “um sistema utilizado para prover informação, qualquer que seja o uso feito dessa informação” e sistemas de informação gerencial, que seriam todos os sistemas destinados a satisfazer as necessidades dos sistemas gerenciais de uma empresa.

Schroeder e Rodrigues (2002) registram que os sistemas de informação trouxeram para as organizações um novo ciclo tecnológico juntamente com uma promessa de colocar sob o controle do computador cada um dos processos da organização. Isso causa a necessidade do aumento de informação em função do crescimento da empresa e um impacto maior destas informações na tomada de decisão por parte dos executivos à medida que o sistema de informação se torna mais confiável. (OLIVEIRA, 2004, p.48).

Apresenta-se assim o relacionamento da informação com o processo de administração dentro das organizações, explicando-se que a administração pode ser compreendida como sendo um processo que transforma a informação em ação, bem como a existência de uma realimentação da ação com uma nova informação, de forma a se transformar num ciclo fechado e contínuo. A Figura 5 detalha o relacionamento informação – decisão - ação.

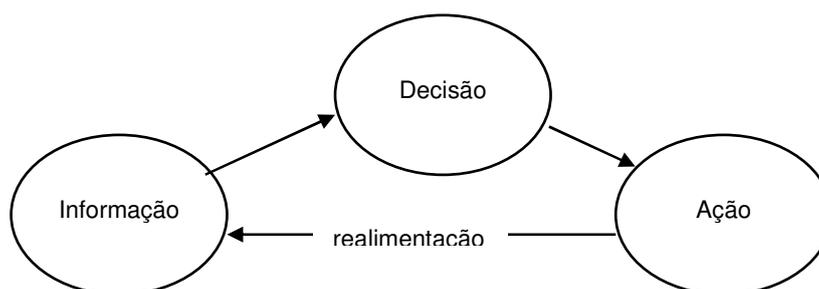


Figura 5 – Relacionamento: informação – decisão – ação

Fonte: Cautella e Polloni (1982, p. 166)

Para Bernardo (1999) os elementos críticos no processo da administração são decisões e pessoas, sendo que o elemento crítico no processo de decisão é a informação. Dentro deste contexto é possível compreender a importância e a valorização dada à administração da informação, uma vez que a gestão baseada na informação é a base do paradigma da gestão moderna que visa a coleta, o processamento e a transformação de dados em informação e sua distribuição.

2.3 Histórico dos sistemas de gestão empresarial

Em uma visão contemporânea, Laudon e Laudon (2001), comentam o fato que um SIG é hoje um sistema multidisciplinar que não é dominado por uma teoria única, justamente pelo grande alcance de aplicação a que chegou o sistema, atingindo dimensões técnicas e comportamentais. Os autores descrevem o SIG como um sistema sócio-tecnológico e apresentam como participantes da abordagem técnica as áreas de ciência da administração, ciência da computação e pesquisa operacional. Como integrantes da

abordagem comportamental são incluídas as áreas de psicologia, economia e sociologia. Esta visão é consequência de uma evolução.

Laudon e Laudon (2001, p.10) registram que em 1950 já era possível utilizar um sistema de folha de pagamento computadorizado. Davenport (2002, p.24) é mais exato ao datar como 1954 o ano de desenvolvimento do primeiro aplicativo para fins de negócio. A descrição de Davenport confere com a de Laudon e Laudon ao descreverem que o aplicativo era uma folha de pagamento que foi desenvolvida pela atual *Andersen Consulting* para a *General Electric*.

Nessa época, as mudanças trazidas pelos aplicativos eram, em sua maioria, na parte técnica. Simplesmente automatizava-se um procedimento. Para Laudon e Laudon (2001), o sistema de informação como controle gerencial começou a existir a partir da década de 60.

A Figura 6 demonstra a mudança do enfoque dos sistemas de informação da década de 50 até a década de 90.

Sistemas de informação	Período	Enfoque
	1950	Mudanças técnicas
	1960 – 1970	Controle Gerencial
	1980 - 1990	Atividades institucionais essenciais

Figura 6 – Enfoque dos sistemas de informação ao longo do tempo

Fonte: Adaptado de Laudon e Laudon (2001, p. 10)

À medida que o enfoque do sistema de informação foi se alterando diferentes siglas e definições foram usadas. Este fato é relatado nas subseções a seguir.

2.3.1 MRP - Material Requirement Planning

Os primeiros MRP datam do final da década de 60, mas é na década de 70 que sua utilização cresceu, conforme descrevem Ritzman, Krajewshi e Moura (1989). Nesse período, o foco do sistema era apenas o Planejamento das Necessidades de Materiais (*Material Requirement Planning*), chamado também de MRP ou MRP I.

O MRP foi um sistema criado com o objetivo de controlar o estoque e dar apoio às funções de planejamento da produção e compras (COLANGELO FILHO, 2001). Os objetivos do MRP são um pouco mais detalhados por Corrêa e Giansesi (1996) quando definem o objetivo dos sistemas de cálculo das necessidades de materiais como sendo permitir o cumprimento dos prazos de entrega dos pedidos do cliente com mínima formação de estoque, fazendo o planejamento de compras e de produção para que ocorram nas quantidades e nos momentos corretos.

Dentro desses objetivos, Ritzman, Krajewshi e Moura (1989) apontam como as metas mais importantes a manutenção de um baixo nível de estoques, e as atividades de atendimento ao cliente, que estão diretamente ligadas ao nível de controle de estoques, uma vez que estas são as duas metas mais visíveis traçadas pela Administração de Materiais.

Ritzman, Krajewshi e Moura (1989) registram também como outro importante objetivo do MRP a produtividade da mão-de-obra, já que o controle de materiais, tempo de preparação e quebra de máquina influenciam diretamente no tempo de preparação de máquinas. Aqui a questão é reduzir o tempo de preparação, aumentando a produtividade.

Outra área afetada é o custo de materiais que é determinado pelas informações sobre o estoque, que influenciam as decisões tomadas pela Administração de Materiais sobre quando comprar e quanto comprar.

Outros aspectos incorporados ao MRP são a Previsão de vendas e a Liberação de Ordens. A Previsão está relacionada com o tipo de produção adotado, se é sob encomenda ou para estoque e a Liberação de Ordens ocorre quando é necessário disparar uma nova ordem de compra ou de produção. Quanto à Liberação de Ordens, deve existir condições de realizar toda a parte de manutenção nessas ordens, ou seja, poder cancelar, alterar, inserir e consultar tais ordens.

Existem mais duas atividades envolvidas: a manutenção de registros e a coordenação. A manutenção de registros deve existir para poder garantir a acuracidade dos registros, visto que uma quantidade de estoque registrada de forma errônea pode, por exemplo, alterar todo o sistema MRP.

Quanto à atividade de coordenação, ela é intrínseca às atividades organizacionais, pois a Administração de Materiais envolve pessoas com necessidades, objetivos e prioridades conflitantes.

Todas essas atividades servem para o desenvolvimento do Plano Mestre de produção, que determina quanto vai ser produzido em um determinado período de tempo.

O Quadro 2 demonstra os objetivos, metas, funções e atividades do MRP, relacionando as metas e objetivos com as funções e atividades.

Metas e Objetivos do MRP	Funções e Atividades do MRP
<ul style="list-style-type: none"> - Rotatividade de estoques - Atendimento ao cliente - Produtividade da mão-de-obra - Custo do material - Custo de transporte 	<ul style="list-style-type: none"> - Previsão e entrada de pedidos - Plano mestre de produção - Plano geral de produção - Manutenção das ordens (colocação, alteração, cancelamento) - Acompanhamento (recebimento e compras de materiais e controle da produção) - Manutenção dos registros - Coordenação

Quadro 2 – Metas, objetivos, funções e atividades do MRP

Adaptado de Ritzman, Krajewski, Moura (1989), p. 5

De acordo com Colangelo Filho (2001), os procedimentos do MRP baseados em listas de materiais e um plano de produção, de forma geral, não davam suporte ao planejamento de capacidade e custos. Em função das possibilidades oferecidas pelo tipo de hardware existente na época, não havia integração do MRP com outras aplicações usadas pela organização.

Segundo Ritzman, Krajewski, Moura (1989), nessa fase a diretoria tinha seu próprio sistema para desenvolvimento de projeção de vendas, inventários, produção e lucros. Este “plano comercial” não estava interligado aos planos de produção e freqüentemente os planejadores da produção nem mesmo sabiam da existência de um plano comercial. Além disso, a contabilidade possuía seu próprio sistema, assim como as áreas de marketing e distribuição. Foi a partir

da necessidade dessa integração que o MRP expandiu, transformando-se no MRP II. Na próxima seção, o MRP II é apresentado mais detalhadamente.

2.3.2 MRP II - *Manufacturing Resource Planning*

O uso da técnica de cálculo das necessidades de materiais popularizou-se e alguns pesquisadores perceberam que a mesma lógica poderia ser aplicada, com algum esforço adicional, para o planejamento de outros recursos de produção, além dos materiais.

O raciocínio era que se as quantidades de recursos utilizados para a execução de todas as atividades de produção fossem conhecidas, seria possível calcular a quantidade total de recursos necessários ao atendimento dos pedidos e distribuir essas necessidades ao longo do tempo (CORRÊA, GIANESI, 1996).

Na visão de Corrêa e Gianesi (1996) com a extensão do conceito de cálculo de necessidades aos demais recursos de manufatura, o MRP passou a ser chamado de MRP II, e a sigla mudou o significado para uma forma mais abrangente: planejamento dos recursos de manufatura (*manufacturing resources planning*).

Colângelo Filho (2001) situa no tempo a expansão do MRP como acontecida na década de 80, informando que o MRP II passou a englobar além dos recursos de materiais, aspectos financeiros, orçamentos e custeio de produção.

Para Ritzman, Krajewski e Moura (1989) o MRP II pode ser usado também como um simulador, de forma a ajudar encontrar respostas para as perguntas do tipo “e se...”. Este tipo de análise, conhecido como “*What-If*” (O que – Se), torna possível ao usuário fazer mudanças em uma ou mais (dependendo do recurso utilizado) variáveis de entrada e observar as mudanças ocorridas nos resultados. Na observação de Coles e Rowley (1996), este tipo de análise pode auxiliar os tomadores de decisão à responder questões do tipo: o que aconteceria com o lucro, se as vendas sofressem um aumento de cinco por cento?

No caso do MRPI, esta funcionalidade seria utilizada para simular certos planos de ação e previsões de demanda, respondendo questões relacionadas

a necessidades e faltas de mão-de-obra, equipamentos, inventário, prazos de entrega e acúmulo de pedidos.

Mas, o princípio básico do MRPII, definido por Corrêa e Giansi (1996) é o cálculo das necessidades, viabilizado pela utilização do computador. Esse cálculo precisa ser feito, conforme já o era na filosofia do MRP, em questão de quantidades e prazos necessários aos recursos de manufatura, buscando sempre a redução dos níveis de estoque.

Os passos para que o cálculo seja realizado é feito, de forma resumida, obedecendo aos seguintes aspectos:

- a) A partir das necessidades de entrega dos produtos finais, chega-se às quantidades e datas;
- b) Realiza-se o cálculo retroativo, no tempo, das datas em que os processos de produção precisam iniciar e terminar;
- c) Determinam-se quais são os recursos e quantidades necessárias para que cada processo seja executado.

Nesta época (década de 80), já se encontram referências do uso do sistema de informação aliado a um banco de dados único, bem como aos conceitos de integração e modularização dos sistemas (CAUTELA, POLLONI, 1996a)

O MRP II está estruturado sob a forma de *software*, em módulos que em geral têm funções diferentes e mantêm relações entre si. Segundo a descrição de Correa e Giansi (1996) são 5 os módulos principais:

- Módulo de planejamento da produção;
- Módulo de planejamento mestre de produção;
- Módulo de cálculo das necessidades de materiais;
- Módulo de controle de fábrica.

A fase de desenvolvimento do MPR II coincide com o período de adoção dos microcomputadores. Como o custo de utilização desses equipamentos era atrativo, a instalação era freqüentemente feita de forma departamental, fazendo com que a aplicações ainda continuassem sendo utilizadas de forma isolada, a exemplo do MRP, dificultando o suporte completo e integrado do processo de negócios (COLANGELO FILHO, 2001).

2.3.3 ERP e Sistemas Integrados de Gestão Empresarial

Na década de 90, com a globalização criando um ambiente de negócios extremamente competitivo, o MPR II sofreu nova ampliação incorporando aspectos da área de Recursos Humanos e Finanças, prometendo maior agilidade e redução de custos. Com os novos recursos passou a ser chamado de ERP, ainda que alguns prefiram chamá-lo de ERM (*Enterprise Resources Management*), (Gerenciamento dos Recursos Empresariais) em função da grande amplitude que o escopo do sistema atingiu, abrangendo, além do planejamento, as atividades de execução e controle (COLANGELO FILHO, 2001).

No Brasil, o ERP chegou em meados da década de 90, trazido pelas empresas estrangeiras que para cá vieram nessa época. (HABERKORN, 2003).

Três décadas se passaram desde os primeiros MRP até o ERP, e essa evolução no tempo, bem como a abrangência das áreas de atuação do ERP é ilustrada na Figura 7.

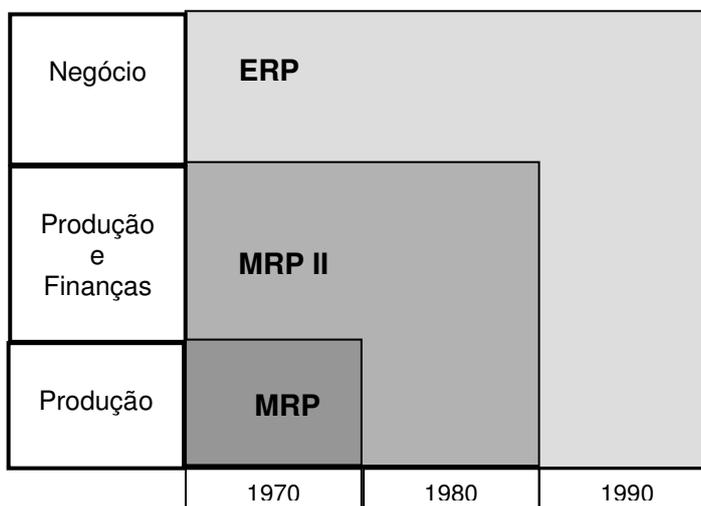


Figura 7 – Evolução dos sistemas ERP

Fonte: Adaptado de Colangelo Filho (2001, p. 21)

Para Slack, Chambers e Johnston (2002, p. 474), o ERP é o “*último e provavelmente o mais significativo desenvolvimento da filosofia MRP básica*”. Apesar de ser fiel ao princípio herdado do MRP de explorar os reflexos que as

mudanças em uma operação produzem em outras, o ERP possibilita que isso seja feito de uma forma muito mais ampla e em todas as partes da organização, refletindo as mudanças de uma parte da operação nos sistemas de planejamento e controle do restante da organização. Isso é possível porque o ERP trabalha com o conceito de ter uma base de dados que integra todas as partes da organização.

O fato da incorporação de novas áreas não dá ao ERP um caráter de *software* para funções estratégicas uma vez que seu foco é voltado para as atividades transacionais do negócio atuando principalmente na questão da integração. As questões estratégicas são abordadas por outros tipos de sistemas de informação como por exemplo: Sistema de Apoio à Decisão, Sistema de Informação para Executivos, *Datawarehouse* e *Business Intelligence*, que farão uso dos dados transacionais coletados com o auxílio do ERP para auxiliar no fornecimento de informações de caráter estratégico.

Sordi (2003, p. 33) relata a inadequação do ERP para funções de ordem estratégica e comenta que a ampla base de dados que o ERP possibilita gerou a evolução e difusão de outros aplicativos como os programas que possibilitam a análise de dados em diferentes dimensões.

Kalpic e Fertalj (2004, p.195) definem com clareza o foco de funcionamento do ERP ao o definirem como um sistema integrado de informação para dar suporte ao negócio nas diferentes partes organizacionais da empresa.

Colangelo Filho (2001) explica que acompanhando os progressos tecnológicos das áreas de hardware e *software*, o ERP é capaz de trabalhar em um ambiente cliente-servidor, com uma base de dados única e integrada.

Jakovljevic (2000) explica que em uma arquitetura cliente-servidor o trabalho é dividido entre os computadores. O “cliente” é o computador que está na mesa (*desktop*) do usuário final do sistema que processa somente algumas funções lógicas e apresenta os resultados processados pelo sistema, enquanto o “servidor” é o computador que contém o banco de dados e os programas aplicativos. O autor explica também que esta arquitetura pode ser implementada de duas formas: em duas camadas ou em três camadas.

Cada camada representa uma parte do sistema como um todo: uma camada para interface, outra para as regras de negócio e uma terceira para o banco de dados.

No conceito de cliente-servidor em duas camadas, a máquina cliente se conecta à um único servidor. Este servidor normalmente controla o banco de dados, enquanto o computador cliente controla a interface do sistema. Neste caso, o servidor se encarrega de cuidar da camada de banco de dados ficando as camadas de interface e da camada de regras de negócios para o cliente.

A Figura 9 apresenta a arquitetura cliente-servidor em duas camadas.

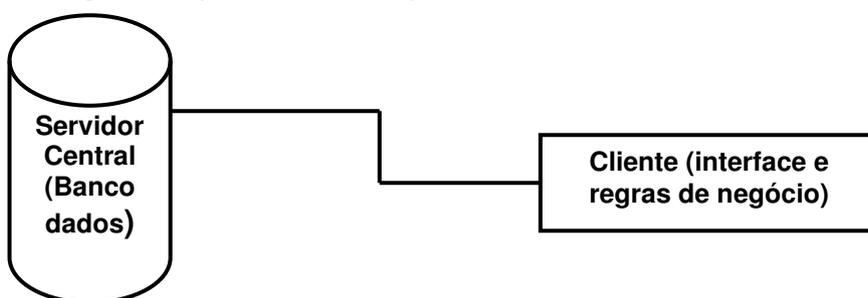


Figura 8 – Arquitetura cliente-servidor em duas camadas

Fonte: adaptado de Jakovljevic (2000)

No cliente-servidor em três camadas a camada de regras de negócios é colocada em um outro computador, ficando o cliente com a interface, outro computador funcionando como servidor de aplicação (regras de negócio) e outro servidor para o banco de dados. Na Figura 10 apresenta-se a estrutura cliente-servidor em três camadas.

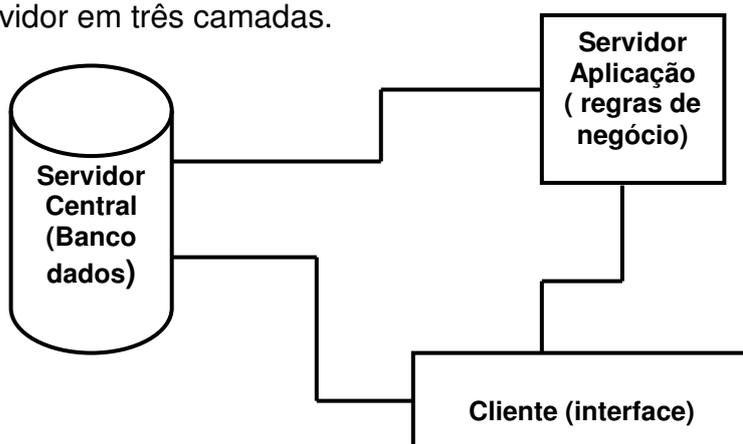


Figura 9 - Arquitetura cliente-servidor em três camadas

Fonte: adaptado de Jakovljevic (2000)

Cada cliente pode utilizar um tipo diferente de aplicação (módulo) do sistema ou um conjunto deles.

Jakovljevic (2000) informa que o padrão cliente-servidor contrasta com o ambiente de funcionamento do MRP que era baseado na utilização de um mini-computador (*mainframe*) ou na utilização do sistema isolada de um microcomputador do tipo PC. Para o autor o ERP está baseado em uma arquitetura distribuída, com gerenciadores de banco de dados conhecidos como “banco de dados relacionais”.

Os bancos de dados relacionais são acessíveis por meio de uma linguagem padrão, a *Structure Query Language* (SQL). Quanto à interface do programa, Jakovljevic (2000) registra que o ERP do ano 2000 faz uso de uma interface gráfica (GUI – *grafical user interface*) explorando recursos de tela de computador que apresenta ícones e uso do *mouse* (*point and click*), fato que não existia nos antigos MRP que utilizavam recursos de tela baseados apenas em caracteres.

Para Colângelo Filho (2001) o modelo cliente-servidor descreve uma forma de relacionamento entre computadores, onde um programa chamado de cliente envia uma solicitação de serviços a outro, denominado servidor. Dessa forma, muitos programas clientes podem compartilhar os serviços disponibilizados pelo programa servidor, podendo ou não fazer parte de uma mesma aplicação. Segundo o autor essa é a forma mais comum de implementação de sistemas ERP.

A Figura 10 mostra diferentes aplicações que utilizam uma base de dados integrada.

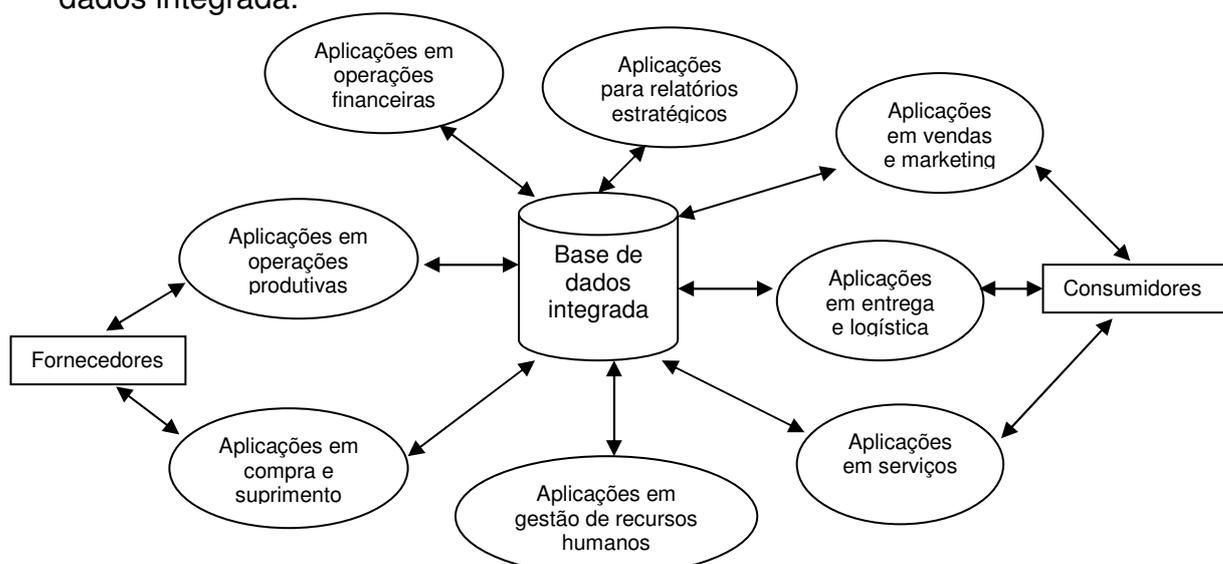


Figura 10 – Aplicações ERP acessam base de dados integrada

Fonte: Slack, Chambers, Johnston (2002, p.474)

Mendes (2003) comenta que os sistemas ERP possuem grande abrangência funcional, objetivando abranger todas as atividades da organização, e fala da divisão do sistema em módulos. Um módulo corresponderia ao menor conjunto de funções que podem ser implementadas separadamente, e estão normalmente ligados a divisões departamentais da organização. Um ponto positivo dos módulos é que eles permitem que a empresa possa implementar somente os módulos de seu interesse, ou implementar o sistema em etapas. De forma genérica os módulos encontrados em sistemas disponíveis no mercado são os seguintes:

- Módulos relacionados a operações e gerenciamento da cadeia de suprimentos;
- Módulos relacionados à gestão financeira, contábil e fiscal;
- Módulos relacionados à gestão de recursos humanos;
- Módulos relacionados ao MRP II (planejamento de materiais, capacidade, gestão de demanda, compras e vendas)

Com o passar do tempo, a sigla ERP pela abrangência de seu significado acabou servindo de classificação para uma determinada categoria de *software*, assim é possível encontrar descrições de programas ditos “ERP para comércio”, ou “ERP para serviços”. Esse fato é comentado por COLANGELO FILHO (2001).

As diferenças entre um tipo e outro estão no objetivo de cada tipo de atividade que o ERP se dispõe a auxiliar, o que irá influenciar diretamente nos módulos que compõem o sistema. Um “ERP para indústria” é o que obedece a descrição histórica, retratando a evolução do MPR e assim terá módulos destinados ao controle de matéria-prima, composição de produtos e cargas de máquinas, entre outros que abrangem funcionalidades de planejamento e controle de produção, módulos que não são necessários por exemplo a um “ERP de comércio”.

Para Davenport (2002), essa categoria de sistema ultrapassou de tal forma o significado literal da sigla ERP que esta deixou de ser a denominação mais adequada, refletindo apenas a origem destes sistemas na área de produção, uma vez que se trata de uma alteração da sigla MRP.

A justificativa para esta visão está no fato de que essa categoria de sistemas deixou de ser aplicada apenas à automação das transações de negócios como por exemplo: atualização de registros contábeis, pagamento de fornecedores, saldo de férias de funcionários, e registros de produção, que necessariamente precisam ser atualizados, mas que eram operações que os clientes não viam e cuja melhoria na exatidão e rapidez não levavam diretamente ao aumento do índice de satisfação do cliente.

Davenport (2002) explica que estas aplicações evoluíram, deixando de estar somente focadas no apoio à retaguarda (*back-office*), e vieram para a linha de frente dos negócios, proporcionando suporte à otimização da cadeia de suprimentos, automação de equipe de vendedores e serviços para consumidores.

Vale dizer que os processos transacionais não perderam a sua importância e que um sistema de informação que apresente deficiências nessa área pode afetar o sistema como um todo, implicando em um mau funcionamento das aplicações voltadas ao suporte ao negócio.

De acordo com Sordi (2003, p.33), os sistemas integrados estão fortemente relacionados com o ERP e acabaram assumindo o significado de um sistema para a estrutura empresarial.

Ainda sobre o uso ou não da sigla, Mendes e Escrivão Filho (2002), registraram informações sobre o que era considerado como característica de um ERP de acordo com quatorze autores diferentes, e apenas três deles consideravam o ERP como uma evolução do MRP. O Quadro 3 registra esse levantamento.

Mendes e Escrivão Filho (2002) concluíram que alguns autores ao caracterizarem o ERP como uma evolução do MRP querem expressar que o embrião do ERP foi o MRP desenvolvido para ambientes de manufatura, mas que a partir disso novos módulos ou mesmo parte de módulos, podem ser introduzidos de forma a adequar o sistema a outros ambientes.

Características	Autores													
	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Auxilia a tomada de decisão			X	X										X
Atende a todas as áreas da empresa	X		X	X	X		X	X	X	X		X	X	
Possui base de dados única e centralizada			X			X			X		X	X	X	X
Possibilita maior controle sobre a empresa			X					X		X	X			X
Evolução do MRP II			X	X					X					
Obtém a informação em tempo real					X					X			X	X
Permite a integração das áreas da empresa	X		X	X	X	X	X		X			X		X
Possui modelos de referência				X					X			X		
É um sistema genérico				X									X	X
Oferece suporte ao planejamento estratégico		X		X	X			X						
Suporta a necessidade de informação das áreas	X	X	X	X	X	X					X	X		
Apóia as operações da empresa	X											X		
É uma ferramenta de mudança organizacional								X		X				
Orientação a processos		X		X		X			X	X	X	X	X	X

Legenda:

01 – Bucknout et.al (1999)

02 – Centola & Zabeu (1999)

03 – Corrêa *et al* (1997)

04 – Cunha (1998)

05 – Davenport (1998)

06 – Deloitte Consulting

07 – Dempsey (1999)

08 – Informática Exame (1997)

09 – Hehn (1999)

10 – Lina et.al (2000)

11 – Mitello (1999)

12 – Souza & Zwicker (2000)

13 – Stamford (2000)

14 – Wood Jr. (1999)

Quadro 3 - Características dos sistemas ERP

Fonte: Mendes e Escrivão Filho (2002, p.281)

Segundo Davenport (2002, p.18), estes sistemas podem ser chamados de “sistemas de gestão empresarial (SGE, ou, ocasionalmente, de ES, da sigla em inglês, para *enterprise systems*)”, bem como são conhecidos como Sistemas Integrados de Gestão (SIG) ou ainda ERP (Enterprise Resource Planning). Todos esses nomes e siglas referem-se a um pacote de programas para dar suporte às necessidades de informação da organização.

Assim, chega-se a denominação de Sistemas de Gestão, ou ainda Sistemas Integrados de Gestão Empresarial (SIGE), como uma referência para o que se chamava de ERP, e em muitos casos as expressões ainda são usadas como sinônimos, ou seja usa-se o ERP referindo-se a sistemas integrados de gestão empresarial.

Toda a questão da mudança de foco e abrangência destes sistemas de informação é que acabou por designá-los como Sistemas Integrados de Gestão, que de acordo com Padilha e Marins (2005), no Brasil são também

conhecidos como de Sistemas Integrados de Gestão Empresarial (SIGE), sendo esta a definição adotada para este trabalho.

2.4 ERP e a era da Internet

A era da internet trouxe modificação também para a estrutura tecnológica dos sistemas integrados de gestão. Sordi (2003, p. 76) relata que os primeiros ERP faziam uso da estrutura proprietária da empresa, que muitas vezes era a mesma que a empresa já possuía quando não havia integração entre os sistemas. Os menus e a interface do sistema também seguiam os modelos dos sistemas anteriores à implantação do sistema integrado.

Com a chegada da Internet e do ambiente *web*, novas possibilidades surgiram e o sistema pode acompanhar esta evolução oferecendo novas opções de configuração.

A versão *web-based* (baseado em *web*) do ERP expandiu os limites de uso do aplicativo. O que antes era restrito aos limites da estrutura da rede de computadores interna da empresa passou a poder ser utilizado a partir de qualquer lugar do mundo que possua um computador com acesso à *internet*.

Para Sordi (2003), isto trouxe facilidades de integração e extensão da rede, além de mudança na interface homem-máquina que passou a fazer uso dos recursos do navegador de páginas *web* (browser), com possibilidades de associação de atividades e até mesmo do usuário do sistema à mensagens de *e-mail*.

Outra modalidade de ERP apontada por Sordi (2003) faz uso do ASP (*Application Service Provider*) ou Provedor de Serviços de Aplicações. Nesta modalidade, uma empresa fornecedora disponibiliza toda a infra-estrutura de computadores, dispositivos de armazenagem de dados, segurança, técnicos e demais elementos da estrutura tecnológica para que o sistema de informação esteja disponível para a empresa cliente que fará uso do sistema por meio de uma conexão de dados, geralmente a Internet.

As empresas fornecedoras de soluções ASP tem procurado alcançar o mercado de empresas de médio porte. A solução do ERP se aplica ao ASP porque, segundo Sordi (2003), trata de processos de negócios que têm uso intensivo e pouca diferenciação entre a maioria das empresas, o que

possibilita gerar uma massa crítica de usuários, com tendência à uma redução nos custos quando comparado ao desenvolvimento ou compra de *software*.

2.5 Abrangência da integração

Tomando por base a questão da integração, a história dos sistemas integrados não tem o ERP como seu último representante. A literatura aponta para novas siglas, que de forma geral, buscam representar a abrangência que a integração assume, desde a integração da manufatura até a integração entre diferentes empresas.

Para Spremic e Vulksic (2005, p. 13), a evolução histórica do ERP divide-se em quatro fases:

- integração de manufatura (MRP);
- integração da empresa (ERP);
- integração com clientes (CRP);
- integração entre empresas (XRP).

As duas primeiras fases já foram tratadas nas seções anteriores deste capítulo, sendo que a primeira fase se refere à integração da programação da produção à ordens de fabricação. A segunda fase se dá quando são integradas as funções chamadas de “*back-office*”, como finanças, controle de qualidade, recursos humanos com objetivo de integrar as necessidades dos processos de negócio da empresa.

A terceira fase, de acordo com Spremic e Vulksic (2005, p. 13), se dá em função da necessidade da integração do *E-commerce* ao ERP. Até então, o sistema ERP era construído para atender situações de produção para estoque (*make-to-stock*), mas, com a integração ao *E-commerce* o sistema precisou ser adaptado para permitir situações de produção de acordo com um pedido específico do cliente (*build-to-order*). Esta é a fase do *Customer-centric Resource Planning* (CRP).

Em sua quarta fase de integração, datada segundo Spremic e Vulksic (2005, p. 13) no início dos anos 2000, as empresas passaram a trabalhar de forma interconectada, criando a necessidade de sistemas globais de informação e surgindo a fase *inter-enterprise Integration* (integração entre empresas) ou XRP (*eXtended Resource Planning*) para suportar a integração

das atividades de negócio internas e externas à empresa como clientes e fornecedores.

À medida que a tecnologia evolui e fornece novas possibilidades de integração, os sistemas integrados fazem uso destas possibilidades e concretizam sua função integradora, ampliando sua abrangência.

2.6 Estratégia, vantagem competitiva e tecnologia de informação

Antes de tratar da presença da tecnologia de informação e dos sistemas de informação no ambiente da pequena empresa, é necessário estabelecer o motivo do uso e preocupação das empresas em relação à utilização dessas ferramentas, bem como as definições de estratégia, vantagem competitiva e diferenciação entre tecnologia de informação e sistema de informação.

Os termos tecnologia de informação e sistema de informação, conforme comentam Ward e Peppard (2002, p.13), apesar de serem muitas vezes utilizados como sinônimos, têm significados distintos.

Segundo Ward e Peppard (2002), a união dos termos “tecnologia de comunicação” e “informação” é geralmente determinado pela sigla TI (Tecnologia de Informação) e designa a convergência destas duas áreas distintas. A tecnologia de informação refere-se especificamente à tecnologia, essencialmente hardware, *software* e redes de telecomunicação, incluindo os outros aspectos tangíveis como servidores, computadores, roteadores e cabos de rede e também os aspectos intangíveis como *software* de todos os tipos.

Ward e Peppard (2002) registram que o termo sistema de informação está ligado à forma como as pessoas e organizações utilizam a tecnologia para coletar, processar, armazenar, utilizar e difundir a informação.

A abrangência do sistema de informação envolve o estudo das teorias e práticas relacionadas com o fenômeno tecnológico e social determinado pelo desenvolvimento, uso e efeito dos sistemas de informação nas organizações e na sociedade.

A conclusão é que tecnologia de informação e sistema de informação se complementam e justamente por isso surge a dificuldade de separá-las e defini-las como coisas distintas. Resta ainda estabelecer como podem interagir com a estratégia de uma empresa.

Rezende e Abreu (2000) estabelecem este vínculo comentando que, no momento em que se decide falar sobre informação empresarial, o assunto passa necessariamente pela discussão do sistema de informação e torna-se necessário discutir estratégias empresariais.

A importância da informação em relação à estratégia é que ela representa o elemento responsável pela integração da estratégia com a ação, fornecendo um retorno necessário à verificação do desempenho da execução em relação ao que foi planejado e dando condições para que seja iniciado um processo de aprendizagem organizacional. Assim, a vantagem estratégica ocorre do resultado do gerenciamento e uso da informação processada pela tecnologia (MCGEE E PRUSAK,1994).

Rezende e Abreu (2000, p.155), ao tratarem de estratégia, fazem uma analogia entre estratégia e guerra; tática e batalha. Isto para esclarecer o conceito de estratégia, pois, segundo Rezende e Abreu (2000), a palavra estratégia está ligada a objetivos macros; ações realizadas de forma mais abrangente, em um prazo e amplitudes maiores, podendo por isso ser decomposta em diversas táticas, que por sua vez tem metas e objetivos mais definidos com ações menores, direcionadas, em prazo e amplitudes menores mas executadas em função de satisfazer a estratégia.

Henderson (1989), ao descrever a origem da estratégia, inicia sua explicação baseando-se no *Princípio da Exclusão Competitiva de Gause*, o qual determina que quando duas espécies competem da mesma forma, por alguns recursos essenciais, mais cedo ou mais tarde uma acaba tomando o lugar da outra. A conclusão deste princípio é que na ausência de forças que possam manter um equilíbrio estável dando à cada espécie uma vantagem em seu próprio território, somente uma delas sobreviverá. Desta forma, a estratégia aparece diretamente ligada à questão da concorrência no ambiente dos negócios.

Porter (1979, p.12) apresenta um modelo para se analisar as forças que governam a competição em um ambiente industrial. O modelo tornou-se referência para vários outros autores, entre eles Stair e Reynolds (1999, p.45), que apresentam os cinco fatores que conduzem a uma vantagem competitiva: a rivalidade entre os concorrentes atuais; a ameaça de novos concorrentes; a ameaça dos produtos e serviços substitutos; o poder de barganha dos

compradores e o poder de barganha dos fornecedores. Por meio da Figura 11, é possível identificar esses fatores e compreender de que forma eles afetam a competição entre as organizações existentes.

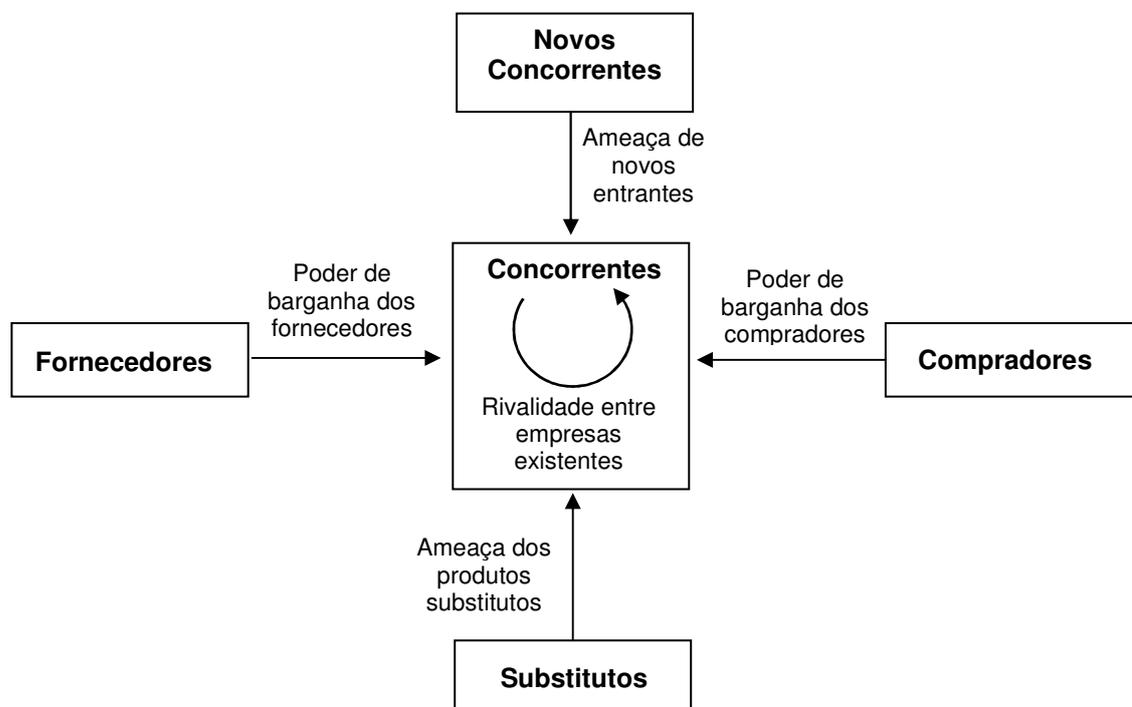


Figura 11 – Forças que dirigem a competição
Adaptado de Porter (1979, p. 12)

Henderson (1989), não distante dos conceitos de Porter, escreve que a estratégia, em se tratando de negócios, é a procura deliberada de um plano de ação que desenvolva uma vantagem competitiva para o negócio. Para o autor, esta estratégia está alicerçada e pode ser colocada em prática baseando-se em cinco elementos básicos:

- habilidade para compreender o comportamento competitivo como um sistema no qual competidores, clientes, dinheiro, pessoas e recursos interagem continuamente;
- habilidade para usar esta compreensão para prever como uma estratégia pode afetar o equilíbrio competitivo;
- conhecer os recursos que podem ser permanentemente utilizados para que os benefícios sejam prorrogados;
- habilidade para prever riscos e retornos com acuracidade e confiança suficientes para justificar um compromisso;
- disposição para agir.

Com bases nestes elementos, cada competidor procurará determinar um caminho para que, a partir de suas capacidades, as diferenças (vantagens) sejam estabelecidas. De acordo com este cenário, Porter (1980, p.35) apresenta três caminhos principais que podem ser escolhidos para intensificar as vantagens, denominando-os de estratégias genéricas. São eles:

1. Custos baixos;
2. Diferenciação de produtos e serviços;
3. Foco no mercado.

McFarlan (1984) faz um detalhamento destas três estratégias mencionando que a redução de custos, por exemplo, é utilizada quando a empresa pode produzir a um custo mais baixo do que seus concorrentes, enquanto que a diferenciação de produtos ocorre quando a empresa oferece produtos com características específicas (como serviços ou qualidade). O terceiro tipo acontece quando a empresa escolhe um nicho de mercado e se especializa em atendê-lo, por isso é chamada de “foco no mercado”.

A partir desses fatores (ou forças), Stair e Reynolds (1999) comentam que uma empresa precisa ser rápida, flexível, inovadora, produtiva e orientada para o cliente para que possa se tornar competitiva. Star e Reynolds (1999) também concordam com McFarlan (1984) ao comentar que diferentes estratégias podem ser adotadas em conjunto para se alcançar a vantagem competitiva, procurando, por exemplo, oferecer um produto ao mesmo tempo diferenciado e a um baixo custo.

McFarlan (1984) aponta como objetivos iniciais do uso de computadores na busca de vantagem competitiva, atender aos objetivos da primeira possibilidade estratégica, ou seja, a redução de custos. Esta redução de custos viria, neste caso, acompanhada de uma maior flexibilidade para os clientes em relação ao tempo e processos.

Segundo Torres (1995), a tecnologia de informação pode orientar uma organização em termos de impactos estratégicos sobre os seguintes aspectos:

- **informática de eficiência:** quando a utilização objetiva reduzir custos, tempos e melhoria de qualidade com relação às entradas e saídas processados pelo sistema (organização) .

- **informática de sinergia e integração:** ao se utilizar a TI para estreitar relações entre a organização e seus cliente e/ou fornecedores.
- **informática de adição de valor ao produto ou serviço:** ocorre quando a própria tecnologia é incorporada ao produto, como por exemplo no caso de enciclopédias multimídia.

Sobre a utilização de sistemas de informação para auxílio à obtenção de vantagem competitiva, Star e Reynolds (1999) comentam que as primeiras aplicações de SI se preocupavam apenas com a questão de redução de custos e melhorias no processamento dos dados das áreas financeiras e contábil, o que está de acordo com o histórico dos sistemas integrados de gestão e as descrições de McFarlan (1984), mas Star e Reynolds (1999) continuam escrevendo que com o passar do tempo, as organizações amadureceram quanto ao uso de sistemas de informação e os gerentes perceberam que poderiam utilizar o potencial dos sistemas de informação e as novas tecnologias para auxiliar a conquista da vantagem competitiva. O cenário encontrado a partir desta constatação foi o de que cada vez mais as organizações utilizam computadores e sistemas de informação para conseguí-la.

Davenport (2002, p.22) apresenta benefícios que as organizações podem obter com a utilização de sistemas de informação, enfocando o uso de Sistemas de Gestão Empresarial (SGE): redução de tempo de ciclo, informações mais rápidas sobre transações, melhoria na gerência financeira, criação de estrutura para implantação de comércio eletrônico e conversão de conhecimento tácito em conhecimento explícito.

Existe ainda um fator de grande importância quando se trata da utilização de sistema de informação como vantagem competitiva que é a interação do sistema com os objetivos, os processos de negócios e as regras de negócio além das próprias pessoas que fazem parte do sistema organizacional. Estes aspectos compõem o que a empresa realmente é e onde ela objetiva chegar após determinada sua estratégia. Caso não haja esta interação, tanto a tecnologia como o sistema de informação podem não responder adequadamente às necessidades da empresa e tornarem-se empecilhos à vantagem pretendida pela empresa.

Sobre este aspecto, Campos e Cazarini (2005b) apresentam um modelo que demonstra que são as características do sistema organizacional (negócio), que darão origem aos objetivos, regras de negócio e processos de negócio que determinam as necessidades e mudanças no sistema de informação e na tecnologia, uma vez que estas devem estar integradas à empresa de forma que possam ser realmente um auxílio à busca da vantagem competitiva.

A necessidade de integração entre estratégia e sistema de informação é tão importante quanto complexa e abrange características ainda não totalmente definidas como, por exemplo, as formas de se medir o quanto esta integração é completa e o valor de seus benefícios diretos e indiretos.

A Figura 12 demonstra o comportamento esperado do sistema de informação, tecnologia e pessoas que compõem o sistema empresa, inserido em um sistema maior (ambiente). As setas indicam que alterações originadas a partir do negócio geram necessidades e adaptações que atingem as diferentes camadas do sistema empresa sob uma visão sócio-técnica (sistema de informação, tecnologia e pessoas).

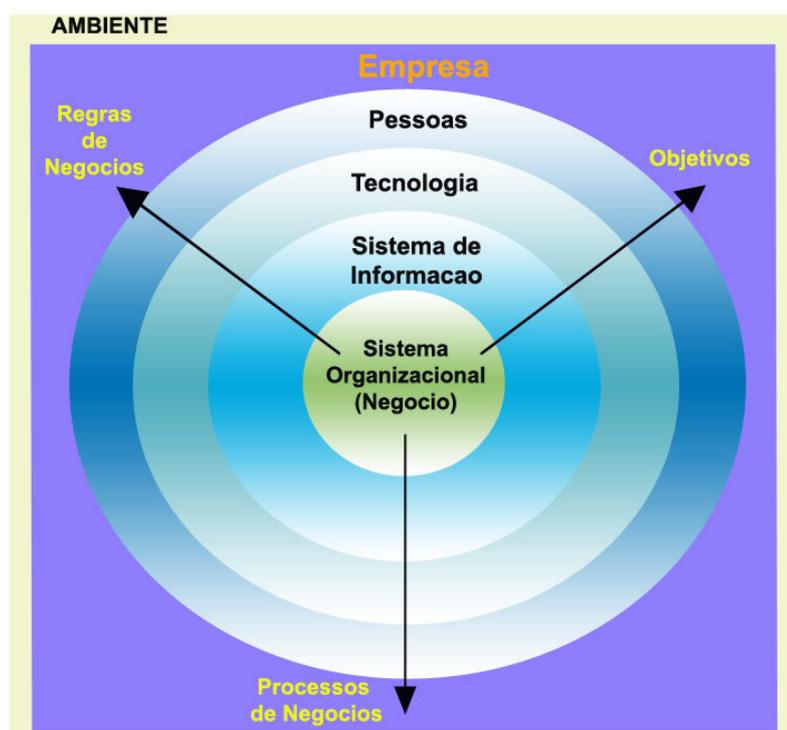


Figura 12 – Integração: organização, sistema de informação e estratégia de negócios

Fonte: Adaptado de Campos e Cazarini (2005b)

A tecnologia por si só não é capaz de produzir todos os benefícios que se espera dela, a menos que seja utilizada com o intuito de servir e integrar-se totalmente à empresa.

Este capítulo apresentou conceitos sobre os sistemas integrados de gestão e de que forma estes podem ser utilizados na busca pela vantagem competitiva, ressaltando a necessidade de integração não só do próprio sistema, mas deste com a empresa. O próximo capítulo tratará de *software* livre.

Capítulo 3. *Software* Livre e *Software* de Código aberto

Apesar dos termos “*Software* Livre” e “Código Aberto” muitas vezes serem utilizados como sinônimos, conforme retrata a Fundação para o *Software* Livre na Europa, FSFE (2004), eles têm significados diferentes. O primeiro, de forma mais abrangente, trata de questões mais filosóficas, preocupando-se com conseqüências econômicas, políticas e sociais que envolvem o relacionamento da liberdade e a existência de *software* proprietário, considerando o fato do código fonte ser distribuído junto com o programa apenas parte das condições para que o *software* livre possa existir. O segundo preocupa-se com as maneiras de garantir de que o código fonte será disponível aquele que quiser utilizá-lo.

Este capítulo apresenta definições, históricos, aspectos legais e sociais do *software* livre. Estes conceitos são importantes para a compreensão do funcionamento, vantagens e desvantagens da utilização do *software* livre.

3.1 *Software* Livre

Tendo como principal órgão controlador a *Free Software Foundation* (FSF), o *software* livre surgiu com o objetivo de desenvolver um sistema operacional que fosse portátil, livre e compatível com o Unix.

No início, os esforços se concentraram em construir os programas que faziam parte exclusivamente do sistema operacional, mas esse objetivo já foi superado, e é possível encontrar diversos tipos de aplicações construídas nos moldes do *software* livre: sistemas gerenciadores de banco de dados, manipulação de imagens e gráficos, servidores de páginas *web*, editores de texto, planilhas eletrônicas, linguagens de programação, gerenciadores de projetos e gestão empresarial entre outros.

3.1.1 Histórico: Projeto GNU, *Free Software Foundation* e Linux

Perens (1999, p.172) escreve que o conceito de *Software Livre* não é alguma coisa nova. Para Stallman (1999, p. 53), o *Software Livre* existe desde o início do desenvolvimento da informática, embora o termo “*free software*” ainda não existisse.

Perens (1999) descreve que quando os primeiros computadores chegaram às universidades, eles eram ferramentas de pesquisa. O *software* era livremente compartilhado e os programadores eram pagos pelo ato de programar e não por seus programas. Somente quando os computadores chegaram ao “mundo dos negócios” é que os direitos ao *software* passaram a ser restringidos e cada cópia passou a ter um custo para ser adquirida.

Em 1971, quando Richard Stallman, fundador do movimento pelo *software* livre, iniciou a sua carreira no MIT (*Massachusetts Institute of Technology*), mesmo as empresas de informática distribuía *software* livre. Programadores eram livres para cooperar entre si, e freqüentemente faziam isso (FSF,2004; STALLMAN, 1999).

Saleh (2004) explica que, nesta época, o foco dos fornecedores de tecnologia era o *hardware*, sendo o sistema operacional muitas vezes oferecido como parte do equipamento uma vez que os aplicativos eram desenvolvidos especificamente para cada arquitetura de computador. Isto fazia com que os programas não tivessem um valor comercial em si e houvesse compartilhamento de código entre participantes de uma mesma comunidade formada por universidades, empresas e instituições de pesquisa. Questões como direitos autorais ou propriedade não eram motivo de preocupação.

Nos anos 80, segundo os registros da FSF (2004), quase todo o *software* passou a ser proprietário, o que significa que ele tinha donos que proibiam e impediam a cooperação entre os usuários.

Na versão de Hohman (1999) e Pereira (2004), Richard M. Stallman era um programador de um laboratório de pesquisa do MIT que queria modificar um *software* de impressão da Xerox para notificar as pessoas do laboratório quando o papel da impressora estivesse preso ou faltando. Mas a Xerox não

liberaria o código fonte, e foi este o fato que levou Stallman a fundar o projeto GNU (GNU's Not Unix) para criar uma opção livre do sistema operacional Unix.

De acordo com Saleh (2004), a história do *Software* Livre como é utilizado atualmente tem seu início com o sistema operacional Unix.

Saleh (2004) conta que o Unix foi desenvolvido nos Estados Unidos pela *American Telephone and Telegraph* (AT&T) em 1969, com o objetivo de gerenciar centrais telefônicas que estavam se tornando maiores e mais complexas. O ponto de partida para o desenvolvimento do Unix foi um sistema operacional anterior chamado Multics, que havia sido criado por uma ação conjunta dos laboratórios Bell, a empresa *General Electric* e o *Massachusetts Institute of Technology* (MIT).

A primeira versão do Unix chamava-se Unics4 e havia sido desenvolvida em uma linguagem de programação chamada "B", mas com o surgimento da linguagem "C", tendo como objetivo ser portátil para diferentes processadores, pois até então os programas precisavam ser escritos para funcionar em diferentes processadores, o Unix foi reescrito em 1973, tornando possível sua adaptação à diferentes arquiteturas de computador.

O Unix agradou os pesquisadores dos laboratórios Bell e várias universidades que se interessaram em contribuir. Como a AT&T, na época, não podia atuar em outro mercado que não fosse o de telecomunicações, cedeu o sistema às universidades e instituições de pesquisa. O resultado foi a implementação de correções de erros e melhorias no sistema, transformando o Unix em um padrão para a área de computação dentro das universidades (SALEH, 2004).

Esta situação permaneceu até 1975, quando o Unix passou a ser controlado, agora com fins lucrativos, pelo *Unix System Laboratories* (USL), uma subsidiária dos laboratórios Bell. Mesmo sob estas circunstâncias a universidade de *Berkeley* não abandonou o projeto do Unix e por meio de um acordo firmado com o departamento de defesa dos Estados Unidos obteve apoio governamental para desenvolver uma versão do Unix: a *Berkeley Software Distribution* (BSD). Porém, ainda não foi suficiente, uma vez que para utilizar o BSD era necessário conseguir uma licença junto à AT&T e USL porque algumas partes do código ainda pertencia a eles. Os custos desta licença se tornaram inviáveis e iniciou-se a tarefa de se reescrever a parte do

código pertencente à AT&T. O resultado foi uma batalha judicial mantida pela AT&T contra *Berkeley* que se estendeu até 1994 quando *Berkeley* pôde finalmente passar a distribuir o *software* sem finalidade comercial.

Enquanto o Unix não se tornava disponível de forma livre, o próprio Stallman (1991) descreve que no ano de 1984 era impossível utilizar um computador sem a instalação de um sistema operacional proprietário e que ninguém tinha permissão para compartilhar *software* livremente com outros usuários de computador, além de que dificilmente alguém poderia mudar o *software* para que satisfizesse suas necessidades particulares.

O Projeto GNU foi fundado para mudar isso tendo como primeiro objetivo desenvolver um sistema operacional portátil compatível com o Unix que fosse 100% *software* livre.

Não 95%, não 99.5%, mas 100% livre - um sistema que os usuários teriam liberdade para redistribuir totalmente, e livres para contribuir e mudarem qualquer parte dele. O nome do sistema, GNU, é um acrônimo recursivo significando "GNU's Not Unix [GNU Não é Unix]" -- uma maneira de pagar o tributo ao Unix, enquanto ao mesmo tempo dizendo que GNU é algo diferente. (Stallman, 1991).

Assim, em 1983, foi idealizado o Projeto GNU como uma forma de trazer de volta o espírito cooperativo que prevalecia na comunidade de informática nos seus primórdios; para tornar a cooperação possível uma vez mais removendo os obstáculos impostos pelos donos do *software* proprietário (FSF, 2004).

Em 1985, foi fundada a FSF – *Free Software Foundation* (Fundação para o *Software* Livre) criada para promover o desenvolvimento e o uso de *software* livre em todas as áreas da computação, mais particularmente para ajudar a desenvolver o sistema operacional GNU e dedicar-se à eliminação de restrições sobre a cópia, redistribuição, entendimento e modificação de programas de computadores (FSF, 2004).

Na definição de Stallman (1991), tecnicamente o GNU é como Unix. Mas diferentemente do Unix, GNU dá liberdade aos seus usuários. Stallman (1991) conta ainda que um sistema operacional não é formado exclusivamente por um único *software*; ele também inclui compiladores, editores, de texto, *software* de *e-mail*, e muitas outras coisas. Então, escrever um sistema operacional completo é uma tarefa muito grande. Desenvolver todos os programas que

compunham o projeto GNU necessitou de muitos anos de trabalho, feito por centenas de programadores.

Apesar de não existir documentado na literatura como se deu o processo de desenvolvimento neste período, Reis (2003, p.17) informa que já existia uma comunidade de desenvolvedores que se interessavam pelos aplicativos e trabalhava para realizar o projeto, utilizando como forma de comunicação a *internet*, que se tornava cada vez mais acessível, correio eletrônico e FTP, embora versões de programas também fossem distribuídos em fita magnética.

Segundo Stallman (1996), no ano de 1992 todos os componentes principais do sistema estavam prontos, exceto o *kernel* (núcleo), que ainda estava sendo escrito. Então o último componente de um sistema similar ao Unix foi desenvolvido: o Linux, um *kernel* livre escrito por Linus Torvalds. Era um núcleo para o sistema compatível com o Unix. Apesar de não ter sido escrito para o projeto GNU, o Linux fez uma combinação útil com o quase completo sistema GNU. Esta combinação disponibilizou todos os principais componentes de um sistema operacional compatível com o Unix, e, com algum trabalho, tornou-se um sistema funcional. Foi um sistema GNU variante baseado no kernel do Linux, e a combinação GNU/ Linux passou a ser usada com sucesso. (STALMAN, 1991).

De acordo com a FSF (2004), a decisão por tornar o sistema operacional compatível com o Unix se deu em virtude de que o mesmo já havia sido testado e era portátil, além de que a compatibilidade tornava fácil para os usuários do Unix a mudança para o GNU.

3.1.2 Definição de *Software* Livre

De acordo com a Fundação para o *Software* Livre (FSF), o conceito de *software* livre está muito mais ligado à condição de “liberdade” que esse tipo de programa proporciona aos seus usuários do que em relação ao fato de se ter que pagar alguma coisa para que seja possível utilizar o *software*.

Conforme as palavras da própria FSF (2004) e Stallman (1999): “*Software* Livre é uma questão de liberdade, não de preço”. Para entender o conceito, deve-se pensar em “liberdade de expressão”, não em “cerveja grátis”.

Para a FSFE (2004), é possível achar que o termo *Free Software* (*Software Livre*) parece ambíguo, já que em inglês, a palavra “*free*” pode possuir dois significados: gratuito e livre. Porém, em várias outras línguas, segundo a FSFE (2004), o termo não apresenta essa ambigüidade, relacionando-se apenas com a liberdade, e que, mesmo no inglês, o problema se resolveria chamando sempre a atenção para o fato de que “*free*”, nesse caso, não tem relação com preço.

O *software* livre se refere, então, à liberdade que os usuários têm de executar, copiar, distribuir, estudar e aperfeiçoar o *software* (FSF, 2003). De acordo com a FSF (2004), Stallman (1999), Perens (1999, p.172) e Saleh (2003, p.27) essa “liberdade” apresenta-se sob quatro formas, que são apresentadas a seguir, numeradas de zero a três:

- 0 - Liberdade de executar o programa, independentemente de qual seja o propósito;
- 1 - Liberdade de poder estudar o programa e modificá-lo, de acordo com as necessidades próprias ;
- 2 - Liberdade de redistribuir cópias do programa;
- 3 - Liberdade de aperfeiçoar o programa, e distribuir os aperfeiçoamentos, beneficiando toda a comunidade.

No conceito da FSFE (2004), a apresentação das quatro liberdades é a forma mais clara para a definição de *software* livre. De forma mais objetiva, Reis (2003, p. 14) define *software* livre da seguinte maneira:

Software livre é qualquer *software* cuja licença garanta ao seu usuário liberdades relacionadas ao uso, alteração e redistribuição. Seu aspecto fundamental é o fato do código-fonte estar livremente disponível para ser lido, estudado ou modificado por qualquer pessoa interessada.

A liberdade de distribuição de cópias, modificadas ou não, pode acontecer cobrando-se ou não uma taxa por isso, bem como não existe necessidade de se pedir ou pagar autorizações ou licenças para que isso aconteça (FSF, 2004).

O fato de se fazer modificações no *software* e utilizá-lo, seja no lazer ou trabalho, também não implica em nenhuma obrigatoriedade, nem mesmo a de

mencionar que essas modificações existem, ainda que o *software* modificado seja redistribuído.

Da mesma forma, a utilização é livre para pessoas físicas ou jurídicas, em qualquer tipo de sistema computacional, para qualquer tipo de atividade.

A distribuição dos programas deve incluir suas formas binárias ou executáveis, bem como o código fonte, seja para as versões originais ou modificadas (FSF, 2004). Isso deve acontecer porque para que a liberdade de estudar, modificar e aperfeiçoar o programa possa acontecer é pré-requisito que se permita o acesso ao código fonte do programa, sendo essa uma condição necessária ao *software* livre.

Reis (2003, p.15) comenta que outras características importantes aparecem ligadas ao *software* livre e este dificilmente está dissociado delas:

- Uso da *internet* para distribuir o *software* e promover o desenvolvimento do *software* de forma descentralizada;
- Os desenvolvedores são também usuários dos produtos que ajudam a desenvolver;
- Existência de interesse pessoal do desenvolvedor (autor);
- Presença de forte individualismo, com desenvolvedores e usuários que se tratam pessoalmente, utilizando nomes próprios e não em termos das organizações que venham a representar;
- Uso de ferramentas de comunicação e desenvolvimento distribuído.

O sentido da definição de *Software Livre* aqui apresentado é aceito tanto pela FSF quando pela *Open Source Initiative*, porém esta definição tem seu foco apenas no produto final gerado pelo modelo de desenvolvimento de *software* livre (REIS,2003). A questão filosófica do movimento e suas conseqüências e características sociais são apresentadas com mais detalhes na seção 3.10.

3.2 Software de Código Aberto

A FSFE(2004) considera o “*Open Source*” como uma campanha de *marketing* para o *software* livre. Com filosofias parecidas, mas objetivos diferentes, os dois movimentos não são conflitantes e podem trabalhar lado a lado em diversos projetos.

O *software* de código aberto também tem um órgão que o orienta: a *Open Source Initiative* (OSI), que tem sua própria história, sua definição e sua forma de aprovação de licenças.

3.2.1 Histórico da *Open Source Initiative*

Para a OSI (2004), a história do “*open source*” mistura-se com toda a história do sistema operacional Unix, a história da *internet* e a história do “*software* livre”.

A definição surgiu em uma reunião, em Palo Alto, Califórnia (EUA), no dia 3 de Fevereiro de 1998. As pessoas presentes na reunião refletiam sobre o anúncio da Netscape de disponibilizar o código fonte de seu navegador para *internet*. Um dos presentes à reunião, Eric Raymond havia sido convidado para auxiliar a Netscape a desenvolver o projeto. Percebeu-se que a Netscape havia criado um espaço precioso para que as pessoas envolvidas com o desenvolvimento de *software* livre pudessem demonstrar os benefícios desta forma de trabalho (OSI, 2004).

O pensamento dessas pessoas era deixar de lado a idéia de “atitude de confronto” que havia sido associada ao *software* livre no passado, e aproveitar a oportunidade dada pela Netscape para vender uma idéia que tinha os mesmos parâmetros do *software* livre. Ao estabelecerem táticas de como isso poderia ser feito, um novo rótulo acabou surgindo, o “*Open Source*”, sugestão de Chris Peterson. A FSFE (2004) classifica o “*Open Source*” como uma campanha de marketing para o *software* livre.

O documento que define o que é o código aberto foi derivado inicialmente do “Guia Debian de *Software* Livre”, escrito originalmente por Bruce Perens. Este guia continha os fundamentos básicos para a distribuição Debian do Linux. O documento original foi refinado e modificado, com a contribuição dos desenvolvedores da Distribuição Debian GNU/Linux, fato ocorrido em junho de 1997.

Em fevereiro de 1998, este documento foi mais uma vez revisado e as referências específicas ao nome Debian foram removidas, tornando-se a definição de *software* de código aberto da *Open Source Initiative* (OSI,2005).

Esta sugestão de definição passou por um processo de discussão entre aqueles que defendiam os ideais do *software* livre e aqueles que entendiam que o melhor seria o código aberto.

Para a FSFE (2004), a terminologia "*Open Source*" refere-se a ter acesso ao código-fonte, mas este acesso é apenas um requisito de duas das quatro liberdades que definem o *software* livre, fazendo com que muitas pessoas não percebam que apenas o acesso ao código-fonte não é suficiente, por isso ainda considera o termo "*Software Livre*" mais abrangente. (Detalhes são apresentados na seção 3.3)

Assim, enquanto o *software* livre está ligado à filosofia de desenvolvimento, referindo-se a questões econômicas, políticas e sociais, indo além da garantia dos direitos de uso e acesso ao código fonte; o código aberto se concentra em determinar regras que possibilitem que o código fonte do *software* o acompanhe quando este for distribuído.

3.2.2 Definição de Código Aberto e seus termos de distribuição

De acordo com Perens (1999, p. 171), a "*Open Source Definition*" é uma lista de direitos para os usuários de computador, definindo certos direitos que uma licença precisa garantir para que seja certificada como *Open Source*. Perens (1999, p.177) escreve que a *Open Source Definition* não tem a intenção de ser um documento legal, nem é por ela mesma uma licença de *software*. É uma especificação de quais são as permissões que precisam existir em uma licença de *software* para que ela seja definida como *Open Source*.

Os termos de distribuição dos programas de código aberto segundo a OSI (2004), precisam estar de acordo com os seguintes critérios:

1 – Distribuição livre: a licença de distribuição não deve restringir a nenhuma das partes interessadas de vender ou ceder o *software* como componente agregado à outra distribuição contendo programas de várias fontes diferentes. A licença não deve cobrar direitos de propriedade ou outras taxas pela venda do programa (SANDER, 2004).

2 – Código Fonte: o programa precisa incluir o código fonte, assim como deve possibilitar a distribuição do código fonte, e da forma

compilada. Quando, por algum motivo, o programa não tiver o código fonte distribuído junto com a aplicação, isso deve estar bem especificado, mas deve existir uma forma de se poder obter uma cópia das fontes via internet, sem custo algum para o usuário. O código fonte deve ser a forma preferida pelo programador para modificar o programa. Não são permitidas formas intermediárias de código, como uma saída do pré-compilador ou tradutor, bem como não é permitido dificultar a compreensão (ofuscar) o código fonte do programa (OSI, 2004).

3 – Trabalhos derivados: de acordo com a OSI (2004), as licenças precisam indicar as modificações e os trabalhos que derivaram do programa original e disponibilizar as licenças sob os mesmos termos da licença do *software* no qual as modificações ou trabalhos derivados se basearam.

4 – Integridade do código fonte do autor: segundo Sander (2004), este critério estabelece que somente é possível restringir a distribuição do código fonte em forma modificada se ela permitir a distribuição de atualizações junto com o código fonte original com o objetivo de modificar o programa durante a instalação. A licença pode também exigir que o programa modificado tenha diferentes nomes ou diferentes versões do *software* original, e também deve permitir explicitamente a distribuição do *software* compilado, a partir do código fonte modificado.

5 – Não discriminar grupos ou pessoas: este critério estabelece que a licença não deve fazer discriminação a nenhuma pessoa, ou grupo de pessoas (OSI, 2004).

6 – Não discriminar campos de interesse: a licença não deve restringir o uso do programa a um campo de interesse específico, seja ele comercial ou não. (SANDER, 2004).

7 – Distribuição da licença: os direitos vinculados ao programa aplicam-se a todos aqueles aos quais o programa é distribuído, sem que seja necessária uma licença adicional para cada um deles. (OSI, 2004).

8 - Licença não deve ser específica para um produto: para a OSI (2004), os direitos vinculados a programas não devem depender dos programas que fazem parte de uma distribuição em particular, ou seja, se os programas foram extraídos a partir de uma distribuição e usados ou distribuídos com os termos da licença do programa, todos aqueles a quem os programas foram redistribuídos devem ter os mesmos direitos concedidos aos que obtiveram o programa originário.

9 – A licença não deve restringir outro *software*: segundo este critério a licença não deve impor restrições a outro *software* que é distribuído juntamente com o *software* licenciado. Por exemplo, não é possível obrigar que todos os aplicativos disponíveis por meio de um determinado *site*, sejam de código aberto porque um dos *software* ali disponível é. (OSI,2004).

10 – Licença deve ser neutra em relação à tecnologia: a OSI (2004), estabelece que a licença não deve estar vinculada a uma tecnologia, estilo ou interface. O objetivo aqui é não permitir que acabe existindo um vínculo entre licença e a forma como o programa é distribuído, por exemplo: permitindo somente a distribuição através de cd-rom, ou criando um determinado *software* que não possa ser modificado para aceitar uma interface que não seja gráfica, ou funcionar fora do ambiente *web*.

3.3 Diferença entre *Software Livre* e *Open Source*

A divisão entre os movimentos de *Software Livre* e *Open Source* tem sido contestada tanto internamente quanto externamente pelos membros da comunidade, mas vale lembrar que a divergência não está nos aplicativos, nos

editores de texto, sistemas operacionais ou compiladores. A divergência está nas licenças (BARR, 2001).

De acordo com FSF (2004), para o *Open Source* o fato do *software* ter seu código aberto é uma questão prática e não ética. O *Open Source* é uma metodologia de desenvolvimento enquanto *Software Livre* é um movimento social. Se, para o *Open Source*, uma solução que não seja *software* livre não é a melhor solução possível, para a *Free Software*, programas que não sejam *software* livre são um problema social e o *software* livre é a única solução.

Open Source e *Software Livre* são como dois movimentos políticos dentro da mesma comunidade. Os movimentos discordam sobre os princípios básicos, mas concordam parcialmente nas recomendações práticas e não se consideram inimigos.

A definição da *Open Source* é muito próxima da *Software Livre*, porém diverge em alguns aspectos que possibilitam a adoção de licenças consideradas restritivas demais pela *Free Software Foundation* (FSF, 2004).

O *Software Livre* têm seus princípios fundamentados nas quatro liberdades já citadas na seção 3.1.2, e considera que o *Open Source* respeita somente duas delas (números 1 e 3), que tornam a presença do código fonte necessária, mas dispensa as outras duas. O quadro 4 apresenta uma comparação entre os movimentos com respeito às liberdades garantidas pelas licenças.

	Usuários têm liberdade para:	<i>Free Software</i>	<i>Open source</i>
0	Usar o programa para qualquer propósito	Obrigatória	Não obrigatória
1	Examinar o código e descobrir sobre seu funcionamento	Obrigatória	Obrigatória
2	Distribuir o programa para outros usuários	Obrigatória	Não obrigatória
3	Modificar o programa	Obrigatória	Obrigatória

Quadro 4 – Liberdades garantidas pelas licenças de *Software Livre* X *Open Source*

Tanto *Software Livre* como *Open Source* utilizam a mesma metodologia de desenvolvimento de *software* (maiores detalhes são apresentados na seção 3.6) , mas com base nas licenças é possível perceber que todo *Software Livre*

é necessariamente também um *Open Source*, mas nem todo *Open Source* é *Software Livre*.

Por uma questão de coerência com os trabalhos acadêmicos de Hexsel (2002), Reis (2003), Saleh (2003) e Augusto (2003) entre outros, em função da indicação da FSF (2003), e pelo fato do foco deste estudo estar principalmente no modelo de desenvolvimento de *software*, onde tanto o termo *software livre* como código aberto (*open source*) são aceitos, este trabalho utiliza o termo *software livre*.

3.4 Categorias de *Software* segundo a FSF

A *Free Software Foundation* também classificou os *softwares* existentes de acordo com algumas categorias. Segundo a FSF (2004), essas categorias podem se sobrepor, ou mesmo fazer partes de outras categorias. A figura 13 demonstra as diferentes categorias de *software* e seus relacionamentos.

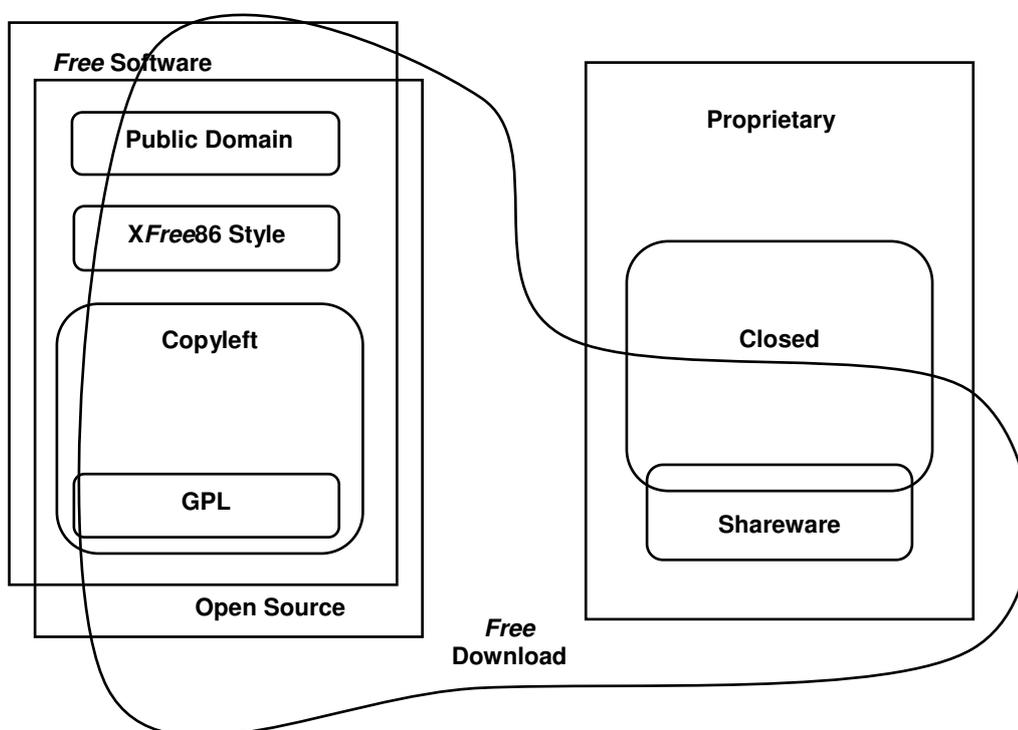


Figura 13 - Categorias de *software* segundo a FSF
Fonte: adaptado de FSF (2004) e Hexsel (2002)

Embora a Figura 13 demonstre as principais categorias de *software* citadas pela FSF, baseando-se conforme explica Reis (2003) na forma como

uma licença protege ou restringe direitos aos usuários, a FSF (2004) descreve algumas outras categorias além das que aparecem na figura. A seguir são apresentadas as categorias da figura 13 e outras descritas pela FSF (2004):

- **Software Livre (*Free software*):** De acordo com a FSF (2003), em uma definição simplificada: o software livre é aqueles que possui permissão para qualquer um copiar, usar e distribuir, com ou sem modificações, gratuitamente ou por um preço.
- **Código aberto (*open source*):** termo usado para designar praticamente o mesmo que *software* livre. A FSF (2003) recomenda o uso do termo *software* livre.
- **Domínio Público:** na definição da FSF (2003), um *software* de domínio público é aquele que não é protegido por direitos autorais e por conseguinte algumas de suas cópias ou modificações podem não ser livres.
- **Software protegido com *copyleft*:** é um *software* livre cujos termos de distribuição não permite que redistribuidores incluam restrições adicionais quando eles redistribuem ou modificam o *software*. De forma prática, isto significa que toda cópia do *software*, mesmo que tenha sido modificada, precisa ser também um *software* livre. (FSF, 2003). Trevisan (2005, p.36) explica que o termo *copyleft* é um jogo de palavras uma vez que o termo *copyright*, utilizado para indicar o direito de impedir a cópia de uma obra artística ou intelectual, é formado pela união das palavras *copy* (cópia) e *right* (direito). Da mesma forma a palavra *copyleft* é uma junção das palavras *copy* (cópia) e *left*, que pode ser traduzido tanto por esquerdo quando como o particípio passado do verbo *leave* (deixar). Assim *copyleft* pode ser traduzido por “deixar copiar”.
- **Software livre não protegido com *copyleft*:** é o tipo de *software* onde o autor permite que além de modificado e redistribuído, pode-se também adicionar restrições ao *software*.
- **Software coberto pela GPL:** de acordo com a FSF (2003), a GNU GPL (*General Public License*) é um conjunto de termos de distribuição

específico para proteger um programa com *copyleft*. A maior parte dos *softwares* do Projeto GNU pertence a esta categoria.

- **Sistema GNU:** sistema operacional livre e completo, no estilo do sistema operacional UNIX.
- **Programas GNU:** equivalente a *software* GNU.
- **Software GNU:** *software* que possui o apoio do Projeto GNU.
- **Software Semi-livre:** para a FSF (2003), um *software* semi-livre é aquele que vem com permissão para indivíduos usarem, copiarem, distribuírem e modificarem desde que não seja utilizado para fins lucrativos.
- **Software proprietário:** é o *software* cujo uso, redistribuição ou modificação é proibido, ou requer permissão para que isso seja feito.
- **Freeware:** segundo a FSF (2003), é o termo usado para *softwares* que permitem redistribuição mas não modificação, pois seu código fonte não está disponível. Reis (2003, p. 13) lembra que *freeware* e *free software* são termos bastante utilizados como sinônimo, mas este uso é incorreto.
- **Shareware:** é o *software* que vem com permissão para redistribuir cópias mas para que se possa utilizá-lo, deve-se pagar por uma licença.
- **Software comercial:** a FSF (2003), define como *software* comercial, aquele que é desenvolvido por uma empresa que procura ter lucro através do uso do *software*. Para a FSF (2003), “Comercial” e “proprietário” não são sinônimos, uma vez que mesmo sendo a maior parte do *software* comercial proprietária, existem *softwares* livres comerciais, e *softwares* não-comerciais e não-livres.

Um aspecto importante a ser notado, conforme lembra Reis (2003, p. 13), é que o termo “*software* livre” inclui o *software* de domínio público, mas não o shareware e o *freeware*.

3.5 Direito autoral e *Copyright*

Hexsel (2002) explica que as licenças de *software* livre fazem uso da legislação de *copyright* para não permitir que os programas sejam utilizados de

forma não-autorizada, mas ao mesmo tempo permitem que estes programas sejam modificados e redistribuídos, o que não é permitido pelo *copyright*. Esta versão de licença com permissões não possíveis à legislação de *copyright* recebeu o nome de *copyleft*. Conforme Bastos et. al (2005), embora o termo *copyleft* não tenha significado legal, o seu sentido é de justamente proibir que haja restrições.

Para a compreensão deste aspecto, uma definição mais detalhada do que é *copyright* se torna necessária e isto esbarra em um outro conceito: o direito autoral.

Losso (2004) e Palermo (2003) explicam que a maioria dos doutrinadores não categorizam o direito autoral como pertencente ao direito da propriedade, mas sim, inserido no âmbito do direito intelectual.

O direito autoral, conforme a definição de Palermo (2003), é a espécie de propriedade intelectual que tutela a proteção da criação e da utilização de obras intelectuais, seja na literatura, artes ou ciência. Para a proteção do direito autoral, Losso (2004) cita três aspectos:

- **O individual:** (direito autoral) baseado na Convenção de Berna, que foi resultado de encontros internacionais realizados entre os anos de 1884 e 1885 na cidade de Berna, na Suíça e assinada em 9 de setembro de 1886. Este sistema é conhecido como “sistema europeu” e é adotado pelo Brasil, onde é fundamentado pelas leis 9.609/98 e 9.610/98.
- **O comercial:** conhecido como “*copyright*”, é o modelo adotado pelos países anglo-saxões. Este modelo concentra-se na reprodução, distribuição, exibição, execução e transformação da obra ou parte dela. Assim está fundamentado na proteção à obra como um direito de propriedade e não aos direitos do autor como criador.
- **O coletivo:** utilizado pela Rússia e pelos países a ela subordinados na época do regime socialista, também é baseado na Convenção de Berna e considerava que a proteção dos direitos do autor deveria servir para a expansão de sua própria cultura (socialista).

Nota-se que embora existam três aspectos para proteção à criação intelectual eles pertencem apenas à dois sistemas, conforme comentário de

Palermo (2003): o europeu e o anglo-saxão. O primeiro de proteção à personalidade do autor e o segundo de proteção à obra.

O direito do autor tem sua origem histórica na França do século XVI, quando os escritores começaram a reivindicar seus direitos na venda e na reprodução de suas obras.

O *copyright* justifica sua tradução literal: “direito de cópia” pois tem sua origem no direito de cópia e comercialização de escritos dado à associação dos donos de papelaria e livreiros da Inglaterra, também no século XVI. Estes comerciantes manipulavam os escritos conforme o interesse real em troca de proteção governamental ao seu domínio de mercado (PALERMO, 2003). De forma mais prática, Bastos et. Al (2005) apresentam a tradução de *copyright* como “direito de uso”.

Apesar da origem histórica do termo na Inglaterra, foi nos Estados Unidos que o *copyright* se estruturou como sistema de proteção intelectual em forma de Lei a partir da Constituição Norte-Americana de 1787.

Palermo (2003) escreve que o sistema *copyright* sempre teve como característica intrínseca seu aspecto econômico referenciando o direito patrimonial expressado no direito de reprodução da obra.

Em 1950, uma segunda Convenção Internacional (a primeira foi a de Berna, em 1886) foi realizada em Genebra com a finalidade de adequar os sistemas de forma a respeitar tanto as obras quanto seus autores.

Conforme apresentado por Bastos et. al (2005, p. 18), *copyright* em sua definição formal é “o único direito legal concedido a um escritor, a um editor, a um compositor, ou a um distribuidor para publicar, produzir, vender ou distribuir um trabalho artístico”.

Assim, o detentor do *copyright* pode designar quem e sob quais condições a obra pode ser copiada, distribuída e comercializada. Enquanto isso, o direito autoral permite que o autor possa deixar sua obra sob um *copyright* podendo determinar ou não as regras de uso e distribuição. A posse do *copyright* também pode não ser do próprio autor da obra e dependendo do estabelecido em contrato o *copyright* pode ser inclusive repassado a outro.

Por exemplo, um escritor que entrega seu livro à uma empresa e concorda em ceder seu *copyright* automaticamente está concordando que a empresa detentora do *copyright* estabeleça de que forma o livro será

impresso, distribuído ou comercializado ou ainda ceder o *copyright* a uma outra editora. Porém, o autor ainda pode reclamar caso seu texto seja utilizado para fins que não foram previamente acordados com a empresa, pois a propriedade intelectual de criação da obra ainda permanece sendo sua. Por isso não se pode dizer que direito autoral e *copyright* sejam a mesma coisa.

O Quadro 5 representa as principais características do direito autoral e do *copyright*.

	Objeto	Atua sobre
Direito Autoral	Pessoa do autor	Propriedade intelectual
<i>Copyright</i>	Obra do autor	Forma de publicar, produzir, vender ou distribuir a obra

Quadro 5 - Principais característica de direito autoral e copyright

Bastos et. al (2005, p. 19) explicam que no caso do *software*, o sistema de *copyright* atribui “donos” ao *software* e estes por sua vez têm o direito de estabelecer quais serão as regras de sua distribuição. É em função deste poder que surgiu o conceito de *software* proprietário, restringindo as formas de distribuição e uso. Do mesmo poder se originou o conceito de *copyleft* do *software* livre que permite que o *software* seja copiado e distribuído.

Esta seção, embora tenha definido os aspectos fundamentais do direito autoral e do *copyright*, é uma representação demasiadamente simplificada da realidade onde os assuntos aqui tratados assumem dimensões de importância internacional e motivam a existência de organizações internacionais e escritórios especializados no assunto para tratar dúvidas e conflitos.

3.6 Regras quanto à distribuição e tipos de licença

O fato de um indivíduo ou organização ter em sua propriedade um *software* livre, por exemplo: um editor de textos, não significa que tenha os direitos de propriedade sobre este *software*.

O *copyright* do programa é de propriedade do detentor destes direitos, mesmo que o *software* tenha sido licenciado sob uma das licenças de *software* livre (PERENS, 1999, p. 181).

As licenças de *software* livre garantem direitos de uso, cópia e distribuição, mas existem outros direitos que obedecem ao uso correto do programa em acordo com as leis de *copyright*.

No *software* livre, são resguardados os direitos do criador e mantidos livres os direitos de uso e conhecimento do *software* (com código fonte). Obedecendo o *copyright* e em conformidade com as principais regras de liberdade, foram atribuídas algumas regras à forma de distribuição e utilização de *software* livre. Estas são chamadas de licenças de uso.

Isto permite, por exemplo, que a forma em que o programa se apresenta quando é distribuído esteja sujeita à uma regra de que não se pode simplesmente decidir por vontade própria como será feita sua redistribuição (compilado, empacotado ou somente o código fonte), exigindo que seja também disponibilizado sob a forma que o desenvolvedor do original escolheu, o que não restringe o direito de escolha em distribuir ou não a versão modificada.

Também é aceitável uma licença que determine que caso seja distribuída uma versão modificada, e o desenvolvedor anterior solicite uma cópia, esta lhe seja enviada.

A FSF (2004), menciona que algumas regras de controle de exportação e sanções comerciais podem limitar a liberdade de distribuição cópias de programas internacionalmente, e que nesses casos a única coisa que os desenvolvedores podem fazer é não utilizá-las em seus programas, de forma que essas restrições não afetem as pessoas fora da jurisdição dos governos que as utilizam.

Existe um tipo de “licença padrão”, criado pela *Free Software Foundation*, chamado de Licença Pública Geral (GPL – General Public Licence), e também uma variante desta licença, a Licença Pública Menos Geral (LGPL – Less General Public Licence). A diferença é que a LGPL permite que o *software* desenvolvido passe a fazer parte de programas proprietários.

Quando um determinado programa está pronto para ser distribuído, e seu desenvolvedor resolve criar uma licença específica para seu *software*, mas deseja que a mesma seja considerada como uma licença de *software* livre, essa licença é submetida à análise da FSF. É comum, nesse caso, que a

licença aprovada tenha seu nome ligado diretamente ao nome do programa ao qual ela se refere, como por exemplo “Licença do X11” ou “Licença da Zlib”.

A FSF(2004) possui uma lista de licenças consideradas como *software* livre, classificadas de acordo com alguns pontos básicos:

- Se ela se qualifica como uma licença de *software* livre;
- Se ela é uma licença com *copyleft*;
- Se ela é compatível com a GNU GPL.
- Se ela causa algum problema prático em particular.

Segundo esses critérios, as licenças foram classificadas pela *Free Software Foundation* em três grupos (uma lista mais detalhada é apresentada no Anexo A):

- Licenças de *Software* Livre compatível com a GPL:
- Licenças de *Software* Livre incompatível com a GPL:
- Licenças que não são de *software* livre:

Cada vez que uma licença é submetida a FSF para uma avaliação que determinará a consideração da licença como sendo de *software* livre ou não, cada um dos critérios sobre liberdade é analisado, e alguma vezes isso pode ser um processo demorado que envolve consulta à advogados e determinação exata do significado das palavras que constam dos termos da licença, o que está sempre levando à procura de melhorar e tornar cada vez mais clara a definição de *software* livre (FSF, 2004).

Da mesma forma que a FSF analisa as licenças a ela encaminhadas para aprová-las como sendo uma licença de *software* livre ou não, existe um programa de aprovação de licenças analisadas pela OSI. As licenças que são compatíveis com as principais licenças de *software* livre (GPL, LGPL, BSD e MIT), são também consideradas licenças de código aberto (OSI, 2004).

Perens (1999, p.185) pede para que não sejam escritas licenças desnecessárias e sempre que possível seja feito uso de uma licença já existente. O autor esclarece que a propagação de muitas licenças diferentes atuam em detrimento do *Open Source* pois fragmentos de um programa não podem ser utilizados por outro programa que tenha uma licença incompatível.

Uma característica importante na determinação da licença é o fato dela ser **permissiva** ou **não-permissiva**.

A licença permissiva é aquela que permite que quando o código-fonte a que ela se refere for utilizado como parte de outro produto, ou mesmo que o código-fonte venha a gerar um novo produto inteiro, este seja redistribuído sem a existência da obrigatoriedade de que seja também um *software* livre. Já a licença não-permissiva exige que a redistribuição do código ou utilização dele em parte de outro produto seja feita também em forma de *software* livre. Algumas licenças não-permissivas exigem que caso o código-fonte seja utilizado em parte de um outro produto este novo produto (inteiro) do qual o código-fonte faz parte deva também ser redistribuído como *software* livre. Lerner e Tirole (2000) utilizam a palavra “viral” para explicar o conceito destas licenças não permissivas, usando-a como metáfora para explicar que este tipo de licença “contamina” o produto do qual faz parte ou que dela se origine.

O Quadro 6 representa um levantamento feito na literatura pesquisada sobre os principais tipos de licenças citadas pelos autores:

Licença \ Autor	Augusto	Behlendorf	FSF	Hexsel	Lerner e Tirole	OSI	Perens	Reis	Saleh
Apache			X				X	X	
BSD		X	X	X		X	X	X	X
Debian			X	X	X				
FDL			X						X
GPL	X	X	X	X	X	X	X	X	X
LGPL	X		X			X	X	X	X
MIT			X			X		X	
Mozilla (MPL)		X	X				X	X	
Netscape (NPL)							X		
Open Source	X		X	X	X				
X.org			X	X			X	X	

Quadro 6 – Principais licenças de *software* livre citadas

Em conformidade com o quadro 6 são descritas as licenças citadas:

GNU GPL: considerada a licença mais importante, é um tipo de licença não-permissiva, o que significa que a redistribuição dos aplicativos sob ela licenciados só é permitida se for mantida a garantia de liberdade para aqueles que fizerem uso da cópia, bem como obriga que versões modificadas do

programa original também sejam livres e distribuídas acompanhadas do código fonte.

GNU LGPL e MPL: são não-permissivas, pois exigem a garantia de liberdade inalterada, mas permitem que o código seja utilizado em outros produtos, e não exigem que este tenha que ser licenciado de forma livre. Caso o código utilizado seja apenas parte de um produto maior e tenha sofrido modificações, estas modificações precisam ser disponibilizadas em forma de código fonte, mas a licença não obriga que o produto como um todo tenha que ter seu código aberto.

NPL: licença não-permissiva desenvolvida pela Netscape quando a empresa resolveu tornar *Open Source* seu navegador de página *web* (Netscape Navigator). A licença dá à Netscape o privilégio de relicenciar as alterações feitas por ela em seu próprio *software*. Esta licença deu origem à MPL (Mozilla Public Licence), que é muito parecida com a NPL, porém sem a cláusula de permissão de relicenciar modificações (PERENS, 1999).

Apache, BSD, MIT e X : são permissivas. Permitem que versões modificadas sejam redistribuídas de forma não-livre. Hexsel (2002) conclui que estas licenças garantem o crédito aos autores do *software* mas não tenta garantir que trabalhos derivados permaneçam como *software* livre. Perens (1999, p. 183) explica que no caso do X, BSD e Apache as licenças diferem bastante da GPL e LGP ao permitirem que se faça praticamente qualquer coisa com o *software* licenciado sob suas características porque estes projetos obtiveram financiamentos do governo do EUA, e uma vez que estes recursos vieram do dinheiro arrecadado com impostos pagos pelos cidadãos, eles obtiveram permissão para fazer uso dele da forma como preferirem.

Debian: em 1995 a Debian, uma organização que tem por objetivo contribuir para a disseminação do Linux celebrou um contrato ("*Debian Social Contract*") entre a organização e a comunidade de usuários de *software* livre. A licença Debian é parte deste contrato e é chamada de DFSG - *Debian Free Software Guidelines* (LERNER e TIROLE, 2000; HEXSEL, 2002). Esta licença é não-permissiva.

Open Source (OSI): A licença do Open Source Initiative, de acordo com Hexsel (2002) e Lerner e Tirole (2000) é derivada da Licença Debian, com as menções à Debian removidas, portanto também é não-permissiva.

FDL - (*Free Documentation Licence*): é um tipo de licença específico para a documentos. Foi escrita originalmente para os manuais dos aplicativos do GNU. De acordo com Saleh (2003, p.30), é uma versão da GPL específica para textos escritos, garantindo que o documento seja livre, e continue livre após modificações. Garante também que partes consideradas fundamentais pelo autor não possam ser modificadas.

As licenças de *software* livre possuem um ponto em comum: o fato de negar todas as garantias referentes ao *software*. O objetivo disto é proteger o proprietário do *software* de qualquer responsabilidade vinculada ao programa. Uma vez que o programa é distribuído sem custos e o autor não tem retorno financeiro com o programa, considerou-se justo que não fosse obrigado a gastar tempo com correções de erros ou arcar com pagamento de seguros. Se aqueles que contribuem com o *Software* Livre perderem o direito de negar as garantias e passarem a ser processados pelos seus programas, eles deixarão de contribuir (PERENS, 1999, p.181).

Vale lembrar que o fato da ausência de garantias não diminui a qualidade do *software* livre, pelo contrário, Hexsel (2002) lembra a excelente qualidade destes aplicativos, por vezes superior à qualidade dos aplicativos comerciais, assim como Saleh (2004, p.115) usa o termo “notavelmente sofisticados” para fazer referências aos aplicativos produzidos sob o modelo do *software* livre.

No Brasil, de acordo com Saleh (2004) existem pelo menos duas iniciativas de construção de licenças: a primeira é atribuída ao advogado Pablo de Camargo Cerdeira que criou uma licença denominada Licença Pública Geral (LPG), pretendendo adequar os princípios da GNU GPL às características das leis brasileiras.

O governo federal é o responsável pela segunda iniciativa, adotando os princípios da Creative Commons GPL e LGPL como as licenças oficiais para o *software* livre produzido pelo governo. O órgão do governo responsável por essa criação foi o Comitê para Implementação de *Software* Livre (SOFTWARELIVRE, 2005).

O Comitê para Implementação de *Software* Livre criou uma licença unindo conceitos da Creative Commons e FSF adaptando suas características às necessidades brasileiras. A Creative Commons é uma organização com

objetivo de aplicar os conceitos em uso nas licenças de *software* livre à outras áreas do conhecimento: músicas, artes plásticas e literatura, procurando dessa forma atuar pela universalização dos bens culturais. (SALEH, 2004).

3.7 Projetos de *Software* Livre

Leme Filho (2003, p.6) informa que a palavra projeto tem sua origem no latim “*projectu*” e significa “lançado para adiante”. Para o autor significado da palavra é uma tarefa ou conjunto de tarefas com um objetivo comum e não está restrito à área de informática, sendo utilizado em diversas outras áreas: projeto de lei, projeto arquitetônico, projeto gráfico. O autor lembra ainda que o trabalho em equipe é a essência do sucesso do projeto.

Para a área de desenvolvimento de *software* o projeto é uma das fases que compõe a metodologia de desenvolvimento de sistemas. É a etapa que ocorre após a análise das necessidades e onde são considerados o sistema operacional, a linguagem de programação, o gerenciador de banco de dados, as regras da interface, enfim o ambiente de produção e funcionamento do software (LEME FILHO, 2003, p.38). A figura 14 mostra as fases de desenvolvimento de software destacando a fase de projeto.

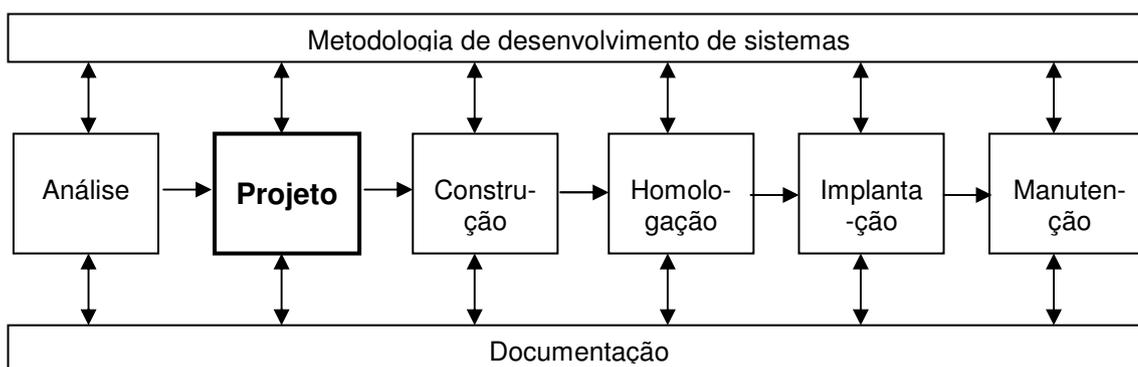


Figura 14 – Etapas de desenvolvimento de software

Fonte: Adaptado de Leme Filho, 2003, p.38

Este trabalho também usa o termo projeto, citando em diversos momentos as palavras “projeto de *software* livre” o que torna importante descrever o que o termo significa para o contexto do trabalho.

Embora segundo Reis (2003, p.21) o termo não tenha uma definição clara na literatura, a definição construída por Reis (2003) será também utilizada neste trabalho:

Um projeto de *software* livre é uma organização composta por um conjunto de pessoas que usa e desenvolve **um único software livre**, contribuindo para uma base comum de código-fonte e conhecimentos. Este grupo terá à sua disposição **ferramentas de comunicação e trabalho colaborativo**, e um conjunto de experiências e opiniões técnicas que usam para discutir o andamento do projeto. (REIS, 2003, p.21).

Esta interação entre usuários em torno da construção de um *software* obedecendo a uma hierarquia não formal, com desenvolvedores atuando também como usuários, e usuários que contribuem apenas utilizando o aplicativo e informando sobre os problemas e os erros cria o ambiente de funcionamento de um projeto de *software* livre. Todas as pessoas que de alguma forma contribuem para os projetos estão inseridas no que é chamado de “comunidade de *software* livre” ou “comunidade *open source*”. A Figura 15 mostra um diagrama que representa os elementos fundamentais um projeto de *software* livre.

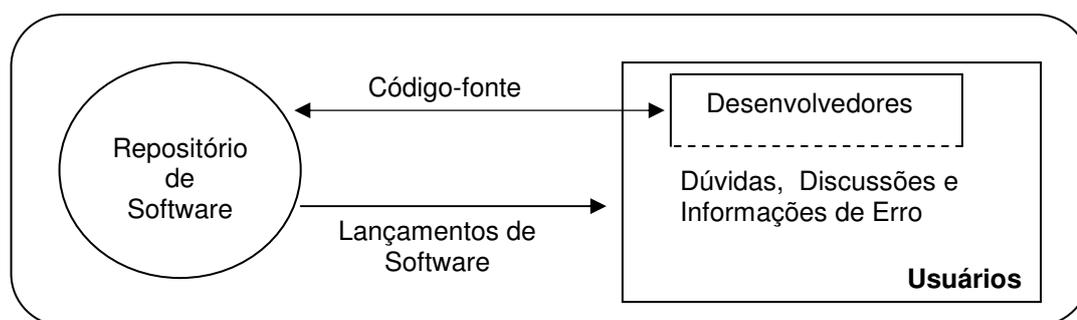


Figura 15 – Diagrama de um projeto de *software* livre

Fonte: Reis (2003, p.15)

Uma vez participante desta comunidade, as pessoas têm sua classificação baseada no tipo de papel que executam. Reis (2003, p.24) relaciona os seguintes papéis:

- **Usuários não-participantes:** a maioria das pessoas faz parte desta categoria. São aqueles que utilizam o *software* como usuário final,

testando-o, mas em geral não informam quando erros acontecem e não têm interesse em discutir ou ajudar no desenvolvimento do *software*.

- **Usuários participantes:** são aqueles que utilizam o programa e informam os erros encontrados e discutem funcionalidades desejadas. Estes contribuem diretamente com o projeto sendo responsáveis pelos testes funcionais.
- **Desenvolvedores esporádicos:** contribuem com alterações localizadas, corrigindo pequenos defeitos ou implementando pequenas melhorias em funcionalidades já existentes. São usuários com conhecimento em programação.
- **Desenvolvedores ativos:** são os responsáveis por módulos inteiros ou pela implementação de funcionalidades mais complexas e extensas. O autor do projeto está entre eles, normalmente como mantenedor central.

Como uma pessoa pode estar ligada a diferentes projetos, ela pode assumir diferentes papéis em cada um dos projetos que participa ou mesmo alterar seus papéis em um único projeto.

O controle das tarefas bem como a coordenação dos projetos é realizado por meio da internet utilizando ferramentas simples e bastante disponíveis, como: correio eletrônico, páginas *web* e listas de discussão. Para a construção do aplicativo são utilizadas ferramentas de desenvolvimento de *software*, controle de versões e acompanhamento de defeitos (REIS, 2003, p.25).

Ainda de acordo com Reis (2003) existe mais 3 formas dos membros da comunidade se organizarem em torno de projetos de *software* livre:

- **Meta projetos:** são vários projetos de *software* livre reunidos para compor um projeto maior.
- **Distribuição:** com o objetivo de garantir simplicidade de instalação, a qualidade do produto e explorar o potencial do mercado, uma distribuição é um conjunto de aplicativos. Diferentemente do meta-projeto que tem por objetivo fomentar e auxiliar os projetos que

engloba, a distribuição tem como meta oferecer uma forma de acesso simplificada ao usuário final.

- **Grupos de usuário:** com objetivo de auxiliar novos usuários a utilizar um projeto de *software* livre, promover o uso de *software* livre na comunidade onde estão inseridos ou dar suporte aos usuário do *software*. Rodrigues et al.(2001) retrata a experiência de um grupo de usuários na Escola de Engenharia de São Carlos formado por estudantes e pesquisadores com o objetivo de promover a utilização do *software* livre.

Alguns projetos são citados na literatura como exemplos positivos do funcionamento do modelo de desenvolvimento adotado no *software* livre. O Quadro 7 apresenta os projetos citados, os autores que os citaram e uma breve descrição do propósito do projeto :

Autor	Augusto	Hexsel	Lener e Tirole	Reis	Saleh
Projeto – Descrição					
Apache – servidor de páginas <i>web</i>	X	X	X	X	X
Bind (<i>Berkley Internet Name Domain</i>) - implementação do protocolo DNS (Database Name Service)	X	X			
Emacs - editor de texto	X	X			
gcc (GNU <i>Compiler Collection</i>) - compilador para linguagens C, C++, Objective C, Fortran, Java e Ada.	X	X			
gdb (GNU Project Debugger) - permite a visualização das atividades de um programa enquanto este é executado.		X			
Gnome – interface gráfica	X				
KDE – interface gráfica	X				X
Linux – núcleo do sistema operacional Linux	X	X		X	X
Mozilla – navegador <i>web</i>				X	
Perl (<i>Practical Extraction and Report Language</i>)- linguagem utilizada para gerar relatórios	X		X		
Sendmail – agente de transporte de <i>e-mails</i> . Também conhecido como servidor de <i>e-mail</i> .	X		X		

Quadro 7 - Exemplos de projetos de *Software* Livre

É claro que não se pode concluir que todos os projetos chegam ao seu objetivo. Alguns deles nem mesmo chegam a sair da idéia inicial de seu

desenvolvedor. Mas Saléh (2003, p.36) cita exemplos de grandes empresas que desenvolvem programas proprietários e que decidiram aderir ao *software* livre em pelo menos um de seus produtos ou têm contribuído no desenvolvimento de outros projetos de *software* livre:

- **Silver Graphics (SGI):** que liberou um sistema de arquivos chamado XFS.
- **IBM e Oracle:** que participam do desenvolvimento do núcleo do Linux em virtude do interesse e utilização de seus produtos neste sistema operacional.
- **Sun Microsystems:** com seu produto StarOffice que acabou originando o projeto OpenOffice (uma suíte de aplicativos contendo editor de texto, planilha eletrônica e gerador de apresentações)
- **Compiere Inc:** que liberou um sistema de gestão empresarial, que recebeu o mesmo nome da empresa, desenvolvido originalmente sob encomenda da Goodyear da Alemanha. O Compiere faz parte dos *software* analisados neste trabalho e é citado também por Appelbe (2003).

Os projetos de *software* podem ser cadastrados nos repositórios de *software* (vide figura 15), providos pelos “serviços de hospedagem de projetos” que oferecem funcionalidades e ferramentas para a manutenção, armazenagem e controle dos projetos. Reis (2003, p.32) descreve as funcionalidades destes serviços de hospedagem destacando:

- Espaço para armazenagem de páginas *web* referente ao projeto;
- Espaço para armazenagem do aplicativo desenvolvido possibilitando o *download* por aqueles que por ele se interessarem;
- Possibilidade de criar listas de discussão;
- Ferramentas de acompanhamento de defeitos;
- Controle de acesso respeitando permissões para diferentes tipos de usuários;
- Formas de publicidade por meio de publicações nos serviços de busca;
- Ferramenta CVS (Concurrent Version System) que possibilita o controle de diferentes versões do aplicativo (normalmente enquanto

uma versão considerada “estável” é liberada para *download*, outras versões estão sendo trabalhadas contemplando implementação de melhorias ou correções de erros).

Como exemplos deste tipo serviços pode-se citar o *site Sourceforge* considerado o maior repositório de projetos e o *site* brasileiro Código Livre.

3.8 Utilização: Vantagens e desvantagens

O uso de *software* livre tem se ampliado, mas alguns setores parecem aceitar melhor seu modelo. Appelbe (2003) cita, por exemplo, o setor de Pesquisa e desenvolvimento em biotecnologia onde o *Open Source* seria mais utilizado que o *software* proprietário.

A maior aceitação por parte de alguns setores se deve ao fato da própria evolução dos aplicativos de *software* livre, que se preocupou inicialmente com aplicativos e ferramentas para desenvolvimento do sistema operacional. Por exemplo: compiladores foram criados oferecendo grande qualidade e flexibilidade, mas os usuários eram conhecedores das funções e da forma de utilização destas ferramentas e dispensaram em um primeiro momento a interface gráfica e algumas facilidades de utilização, o que afastava aqueles que não possuíam conhecimento mais profundo.

Em seguida surgiram aplicativos com direcionamento para aplicações específicas como servidor de páginas *web* (Apache), servidor de *e-mail* (Sendmail), aplicativos para transferência de arquivos (servidor FTP) e aplicativos para facilitar a utilização de máquinas Linux e Windows em uma mesma rede de computadores e o desenvolvimento de ambientes de interface gráfica (Servidor X, KDE, GNOME, WindowMaker). Estas aplicações, aliadas à confiabilidade alcançada principalmente pelo Linux, impulsionaram sua utilização na função de servidor.

Embora a instalação dos aplicativos já tivesse sido facilitada, restava ainda algumas dificuldades com relação à parte de *hardware*, principalmente quando novas tecnologias eram utilizadas. Os desenvolvedores de *software* livre que trabalhavam nos programas (*drives*) que gerenciam os dispositivos de placas de vídeo, *modem*, *mouses* e placas de som que adotavam novas tecnologias muitas vezes não obtinham do fabricante o código fonte para

adaptá-lo aos sistemas operacionais “livres”, nem tampouco o próprio fabricante disponibilizava o *driver* pronto, obrigando o uso de engenharia reversa para que se descobrisse como a nova tecnologia era utilizada pelo dispositivo para depois poder tornar o dispositivo compatível.

A demora na escrita destes *drives* era agravada pelo fato de que os voluntários que trabalhavam em seu desenvolvimento não o faziam em tempo integral. Nessa fase, utilizar um *modem* gerenciado por *software* (*winmodem*) ou mesmo um *mouse* do tipo PS2 era bastante complicado, para não dizer quase impossível.

Com o aumento do número de usuários aliado à participação de grandes empresas (IBM, SUN, HP) nos projetos de *software* livre, os fabricantes passaram a liberar o código ou disponibilizar o *driver* tornando seus produtos compatíveis com o Linux.

Outro fator que contribuiu para o aumento da utilização do *software* livre foi a disponibilidade de aplicativos que se utilizam da interface gráfica, já consagrada pelo Windows®. Os aplicativos StarOffice e OpenOffice entre outros, passaram a oferecer possibilidades de uso de editores de texto e planilhas eletrônicas de uma forma que não era exclusividade de usuários experientes. As facilidades do “click” e do “arrastar e soltar” estavam disponíveis ao usuário comum.

Atualmente é possível encontrar distribuições de Linux brasileiras, como a Kurumim e Kalango, que oferecem, além de aplicativos com interface gráfica, a detecção automática dos dispositivos de *hardware* e ainda funcionam a partir do *cd-rom* sem a necessidade de instalação permanente no computador do usuário, a menos que este assim deseje. É o tipo ideal para aqueles que querem um primeiro contato com este tipo de sistema operacional.

Pode-se notar que, até então, as questões de utilização do *software* livre sob o conceito técnico baseiam-se principalmente em dois aspectos: a funcionalidade e a compatibilidade. Junte-se ao conceito técnico o conceito econômico da não necessidade de pagamento de licença e pode-se discutir uma perspectiva de valor relativo de utilização do *software* livre. Appelle (2003) discute esta perspectiva escrevendo que a mesma é dependente de dois pontos de vista diferentes: o da academia, representado pelas universidades de computação e tecnologia da computação e o das empresas,

representado principalmente pela Microsoft®. Para o autor, pode-se dizer que nenhuma das duas visões é totalmente imparcial, pois, enquanto a Microsoft tem seus interesses comerciais, a academia é em sua maioria a favor do *software* livre uma vez que este reduz o custo de infra-estrutura e soluciona problemas de licenças além de possibilitar experiências com adaptação, construção e produção de *software* que não poderiam ser realizadas com *software* proprietário.

Na opinião de Appeble (2003), em cinco anos não existirá diferença com relação à funcionalidade e compatibilidade entre os produtos *open source* e proprietário. A escolha deixará de ser baseada na funcionalidade e compatibilidade para se basear em preço, suporte e Custo Total de Investimento (TCO), da mesma forma que a questão não será mais quantos usuários experientes ou usuários domésticos utilizam *open source*, mas quantas empresas e negócios fazem uso dele.

3.8.1 Vantagens

Saléh (2003, p.41) apresenta vantagens na utilização de *software* livre e considera como principal elemento impulsionador de sua utilização o fato do não pagamento de licenças. Além da vantagem do baixo custo, outras vantagens precisam ser consideradas e são apresentadas a seguir:

- **Baixo custo econômico:** de acordo com Kon (2001), em geral, o custo de aquisição de *software* aberto é muito baixo quando comparado aos *software* proprietário. Hexsel (2002) chama esse benefício de “desembolso inicial próximo de zero”, considerando que se pode obter o *software* pagando apenas pelo preço da forma como ele é distribuído (cd-rom, ou conexão com internet). Isto se explica pelo modelo de produção do *software* livre que considera, segundo Saléh (2003), o custo principal para produção do *software* é o seu desenvolvimento, já que uma vez produzido, não existe grande custo em sua distribuição. É o mesmo produto distribuído para quantos usuários o quiserem. Sob a visão de baixo custo estão também os valores cobrados pelos *softwares* proprietários quando da necessidade da atualização do *software* para

correção de erros ou aquisição de versões com novas funcionalidades. O *software* livre também não cobra por isso.

- **Baixo custo social:** enquanto o *software* proprietário busca benefícios direcionados para seu fabricante e o lucro das vendas são sempre privados, o *software* livre têm seu benefício focado em seus usuários e os frutos de sua produção tornam-se disponíveis para toda a comunidade (HEXSEL, 2002).
- **Independência de tecnologia proprietária ou de fornecedor único:** *software* livre utiliza padrões abertos de comunicação e armazenamento de arquivos, o que evita o risco de dependência de um único fornecedor caso este venha a criar seu próprio padrão e o mantenha somente sobre seu conhecimento (SALEH, 2004). De acordo com Hexsel (2002) isto se aplica também no caso do fornecedor desistir da fabricação do produto.
- **Robustez:** o modelo de desenvolvimento de *software* livre proporciona testes realizados pelos próprios usuários que garantem a boa qualidade, estabilidade, desempenho e segurança do *software*. Raymond (1998) compara o modelo de desenvolvimento do *software* livre a um produto de bazar com contribuição de muitas pessoas, enquanto o modelo de *software* proprietário seria como a construção de uma catedral, baseado em um grande projeto mantido apenas por especialistas. Como consequência o modelo que mais pessoas contribuem e testam tende a ser melhor do que o que se concentra na opinião de alguns.
- **Garantia contra descontinuidade:** no caso do abandono do desenvolvimento do produto por parte de seus desenvolvedores é possível a contratação de serviços ou assumir seu desenvolvimento (SALEH, 2004).
- **Possibilidade de personalização:** considerada por Hexsel (2002) como uma das maiores vantagens do *software* livre, é a possibilidade de alterar o sistema às necessidades específicas do usuário uma vez que o código fonte acompanha o aplicativo.
- **Não obsolescência do hardware:** de acordo com Hexsel (2002, p.13) é comum que o fornecedor de *software* proprietário ao publicar uma nova versão de seu produto, indique a atualização também do

equipamento que o executa, uma vez que as novas funcionalidades (nem sempre necessárias) requeiram equipamentos com maior capacidade de processamento e armazenamento. Isto é conhecido como “*software bloat*” ou “inchaço do *software*” e acontece em escala muito menor no *software* livre que não sofre a pressão de *marketing* exercida pelos produtores de *software* proprietário.

3.8.2 Desvantagens

De acordo com Hexsel (2002), o *software* livre apresenta também algumas desvantagens que são descritas a seguir:

- **Instalação e configuração difíceis:** o fato dos primeiros usuários do *software* livre serem os próprios programadores ou usuários mais avançados faz com que os critérios de facilidade de instalação e configuração sejam por eles definidos, tornando mais difícil a instalação para usuários menos experientes. Este fato têm mudado à medida que usuários menos experientes têm procurado utilizar estes aplicativos e contribuído com solicitações de funcionalidades de instalação e configuração que para eles são interessantes.
- **Interface de usuário não padronizada:** em consequência do desenvolvimento descentralizado de aplicativos existe a falta de um estilo bem definido e uniforme para formatação da interface de usuário. Esta situação está sendo remediada por projetos que pretendem melhorá-la, como por exemplo o KDE (*K Desktop Environment*) e GNOME (*GNU Object Model Environment*).
- **Mão-de-obra escassa:** existem indicações de que o número de pessoas qualificadas é pequeno frente à demanda tanto para desenvolvimento quanto para a administração de sistemas. As ferramentas para desenvolvimento de sistemas distribuídas como *software* livre, ainda que de grande qualidade, são menos amigáveis e exigem uma maior sofisticação por parte do desenvolvedor, o que ajuda a aumentar o *déficit* de pessoal.

O Quadro 8 apresenta as possíveis vantagens e desvantagens do *software* livre.

Vantagens	Desvantagens
<ul style="list-style-type: none"> ▪ Baixo custo econômico ▪ Baixo custo social ▪ Independência de tecnologia proprietária ▪ Independência de fornecedor único ▪ Robustez ▪ Garantia contra descontinuidade ▪ Personalização ▪ Não obsolescência de hardware 	<ul style="list-style-type: none"> ▪ Instalação e configuração difíceis ▪ Interface não padronizada ▪ Mão-de-obra escassa

Quadro 8 - Vantagens e desvantagens do *software* livre

O aspecto “garantia contra descontinuidade”, apontado como uma vantagem, pode apresentar algumas dificuldades. É certo que um projeto considerado “de sucesso” dificilmente é abandonado, seja pela sua utilidade, seja pelo número de colaboradores dispostos a assumir a responsabilidade caso algum membro que exerça uma função mais estratégica não possa mais desenvolver seu trabalho. Porém simplesmente apontar para a possibilidade de assumir o desenvolvimento ou contratar serviços para que o projeto continue é uma decisão que pode não ser tão simples assim. Pode ser que isso não seja viável quando se considera a hipótese de utilização de um aplicativo em ambiente empresarial. Neste caso, uma série de outras decisões precisam ser tomadas com relação a custo, disponibilidade de tempo e de recursos humanos.

3.8.3 Desvantagens imaginárias

Hexsel (2002, p.18) aponta ainda algumas características chamando-as de desvantagens imaginárias. A descrição de desvantagem imaginária se dá em função destas características serem atribuídas por alguns como desvantagens do *software* livre, mas segundo o autor, elas não são de fato desvantagens. A seguir são apresentadas estas características:

- **Ausência de proprietário e/ou responsável legal:** segundo Hexsel (2002), mesmo os aplicativos com fins comerciais utilizam licenças com cláusulas onde o proprietário se exime de qualquer responsabilidade por

danos ou prejuízos mesmo que estes tenham sido causados com a utilização correta do software e em acordo com a documentação fornecida. Como exemplo o autor apresenta a licença do Windows 98 e explica que este tipo de prática comercial é possível porque os produtos de software é legalmente similar a trabalhos artísticos. Dessa forma não seria desvantagem a características das licenças de *software* livre que se isentam das garantias.

- **Inexistência de suporte por meio do modelo tradicional:** o autor considera como modelo de suporte tradicional aquele oferecido em regime de 24 horas durante os sete dias da semana. Hexsel (2002) relata a existência deste tipo de suporte também por empresas prestadoras de serviços especializadas em *software* livre, principalmente no que se refere ao sistema operacional Linux, além de apontar para o suporte que pode ser conseguido por meio da Internet e de ferramentas de comunicação como *e-mail* e listas de discussão.
- **Instabilidade:** segundo o autor existe uma “falsa crença” de que o *software* livre não pode ter qualidade. Isto estaria vinculado à falta de informação pois de acordo com Hexsel (2002, p. 20) não só boa parte da infra-estrutura da Internet está baseada em *software* livre, assim como a própria Internet só se tornou viável porque seu principal protocolo de comunicação o TCP/IP foi disponibilizado como *software* livre pelo *Computer Science Research Group* da Universidade de Berkeley.
- **Poucos aplicativos comerciais:** de acordo Hexsel (2002) já existe um número considerável de sistemas comerciais disponíveis como *software* livre e de aplicativos proprietários que foram adaptados para funcionar sobre o sistema operacional Linux, como por exemplo: StarOffice, Lótus, Corel Word Perfect, e OpenOffice, banco de dados Oracle e aplicativos empresariais da IBM e SAP. Segundo o autor a variedade e o número de aplicativos do tipo folha de pagamento, contas a pagar e similares ainda é pequeno devido à baixa demanda em função do pouco interesse dos empresários que ainda não conhecem o *software* livre ou o consideram um produto de caráter marginal aplicado aqueles que tem

grande conhecimento ou à área acadêmica, fato que faz com que este seja um dos últimos lugares onde o software livre deverá atingir.

3.9 Software livre e governo

Kon (2001) descreve sua preocupação com a existência de milhões de empresas que atuam em todos os setores da economia que são dependentes de um único fornecedor de *software* e registra que a situação se torna ainda mais séria quando governos baseiam sua máquina administrativa inteira neste mesmo fornecedor (Microsoft©).

Alguns trabalhos como os de Hexsel (2002), Borges e Geyer (2004) e Schwingel (2003) indicam a utilização de *software* livre em instituições governamentais.

Hexsel (2002) lembra que a importância estratégica da utilização de *software* livre no governo é indicada também por estudos realizados na Comunidade Européia e pelo governo dos Estados Unidos, sendo que em ambos os casos as recomendações são de incentivo, financiamento e aumento do uso de *software* livre. Pereira (2004) cita o caso de Portugal, que criou a Associação Nacional para o *Software* Livre (ANSOL). Appeble (2003) comenta recomendações de *software* livre nos governos da Bélgica, Espanha, Bulgária e Costa Rica, além do estado de Oregon, nos Estados Unidos.

O governo brasileiro também indica a preferência pela utilização de *software* livre e inclusive criou um órgão: o Comitê para Implementação de *Software* Livre, criado em 29 de Outubro de 2003 e vinculado ao Instituto Nacional de Tecnologia da Informação para promover esta utilização (SOFTWARELIVRE, 2005).

Dentre os objetivos do Comitê está a efetivação do *software* livre como ferramenta corporativa padrão do governo federal. Segundo SoftwareLivre (2005), o Brasil já utiliza *software* livre nos seguintes órgãos de administração pública: Instituto Nacional de Tecnologia da Informação (ITI), Serviço Federal de Processamento de Dados (SERPRO), Centro de Gestão Estratégica do Conhecimento em Ciência e Tecnologia (CGECon), Agência Brasil - Radiobrás, Ministério da Cultura, Ministério da Defesa, Ministério do Desenvolvimento

Agrário, Ministério das Cidades, Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) .

Além dos órgãos federais, *SoftwareLivre* (2005) indica iniciativas por partes dos Estados da Bahia, Santa Catarina, Paraná, Minas Gerais, Espírito Santo, Mato Grosso do Sul, Rio de Janeiro, Rio Grande do Sul, São Paulo e Distrito Federal na utilização de *software* livre e a existência da ABRASOL - Associação Brasileira de *Software* Livre.

Outra ação do governo federal com relação ao *software* livre deu-se por meio de decreto 5.688 que procurou facilitar a captação de recursos para o programa “PC Para Todos”, que objetiva oferecer um computador pessoal com preço mais acessível à população. A redução de preço, segundo Sandrini (2006), é possível por causa de três fatores: a desvalorização do dólar que barateou a importação de componentes, a redução por parte do governo dos impostos IPI e ICMS para computadores com valor até R\$ 2.500,00 e a utilização de *software* livre (sistema operacional Linux) no lugar do Windows®.

Kon (2001) também comenta que em 2001 foi sancionada uma lei pela prefeitura de Recife – PE que determina a utilização preferencial de *software* livre nos sistemas e equipamentos de informática dos órgãos de sua administração direta e indireta.

3.10 Aspectos sociais

O “movimento de *software* livre”, como é chamado em sua abrangência como um todo, incorpora questões que ultrapassam o foco da produção de *software* e atinge aspectos e natureza política, econômica e de relacionamentos entre pessoas. Alguns estudos começam a surgir abordando estes aspectos, como por exemplo: Kon (2001), Augusto (2003), Trevisan (2005), Rodrigues et al.(2001) e Pereira (2003), apontando para o movimento de *software* livre como uma fonte de reflexão sociológica.

Para Pereira (2003), o movimento de *software* livre é ao mesmo tempo:

- Uma fonte de inovação tecnológica que difere do modelo tradicional de produção e comercialização de tecnologia;
- Um movimento com implicações políticas e ideológicas;
- Um movimento que se auto-intitula “comunitário”;

- Uma rede social que une programadores e usuário de diferentes partes do mundo.

O modelo de inovação tecnológica baseado na partilha do conhecimento, na reestruturação do conceito produtor/consumidor (o usuário é também produtor) e no conceito de progressão tecnológica baseada na implementação de contribuições complementares (*patchwork*) substitui o modelo convencional que tem suas bases em procedimentos restritos aos domínios daquele que o produz, na concorrência entre as empresas e no controle sobre a informação.

A utilização de *software* livre em programas de inclusão digital, principalmente por parte dos governos também é um fator que contribui para reflexos sociais da aplicação e uso deste tipo de aplicativo.

Este capítulo apresentou definições e características do *software* livre sobre diferentes aspectos: conceitos, legislação, tipos de licença, utilização pelos governos e aspectos sociais. O próximo capítulo apresenta o ambiente da tecnologia e sistema de informação nas pequenas empresas.

Capítulo 4 - Pequenas empresas

Esse capítulo trata das características das pequenas empresas a partir dos parâmetros utilizados em sua classificação na Europa, Estados Unidos e Brasil. Com base em dados de pesquisas realizadas por organizações ligadas às empresas, é apresentado um panorama da utilização dos recursos de tecnologia, de *software* de gestão e dos recursos humanos ligados à área de tecnologia de informação no ambiente da pequena empresa, particularmente no estado de São Paulo, com a proposta de reunir informações que possibilitem orientar a análise de informações relacionando as características das pequenas empresas e os requisitos de hardware, *software* adicional e conhecimento necessário para a instalação dos aplicativos *Open Source*.

4.1 Características das pequenas empresas

De acordo com Longenecker, Moore e Petty (1997), especificar qualquer padrão de tamanho para se chegar a uma definição de pequena empresa não é algo fácil de se fazer nem existe um consenso estabelecido, já que se adotam padrões diferentes para propósitos diferentes. Mesmo com relação à adoção de quais são os critérios usados para medir o tamanho, existem diferenças, sendo que alguns deles são relevantes apenas para certas atividades de negócio. Longenecker, Moore e Petty (1997) dão alguns exemplos de critérios:

- Número de empregados;
- Volume de vendas;
- Valor dos ativos;
- Seguro da força de trabalho;
- Volume de depósitos.

De acordo com a Associação Empresarial de Portugal (AEP) (2005), a Comissão Europeia adotou pela primeira vez, em 1996, uma definição formal para as micro, pequenas médias empresas tendo seus próprios critérios para estabelecer e classificá-las.

A Tabela 1 aponta essa classificação com informações válidas a partir de 01 de Janeiro de 2005.

Tabela 1 – Classificação europeia de micro, pequena e média empresa

Categoria	Número de Trabalhadores	Volume de negócios	Balanço Total
Média empresa	< 250	50 milhões de euros	43 milhões de euros
Pequena empresa	< 50	10 milhões de euros	10 milhões de euros
Micro empresa	<10	2 milhões de euros	2 milhões de euros

Fonte: Associação empresarial de Portugal

A United States Small Business Administration (SBA) (2005) também tem seus critérios de identificação para as pequenas empresas dos Estados Unidos da América (EUA), baseando-se no tipo de atividade da empresa e a partir desse parâmetro estabelecendo critérios de valor de faturamento ou número de empregados.

A SBA (2005) tem uma tabela de padrões detalhada sobre cada atividade, e uma outra resumida, em que as empresas estão agrupadas por ramos de atividade. Cada atividade recebe um valor diferente, seja em número de empregados, no caso das atividades de manufatura ou comércio varejista seja em valor de faturamento nas demais atividades, que em alguns casos recebem valores diferentes quando é possível uma subclassificação dentro da atividade principal. Um exemplo é o caso de serviços e negócios pessoais que recebe limites diferentes quando os serviços se destinam a arquitetura ou serviços de limpeza de carpete.

A Tabela 2 mostra algumas das atividades encontradas na tabela resumida da SBA com informações que estão em vigor a partir de 05 de Dezembro de 2005.

Tabela 2 – Parâmetros de Pequenas empresas segundo a SBA

Grupo Industrial	Tamanho Padrão (Qtde empregados / Faturamento)
Manufatura	500 empregados
Comércio Atacadista	100 empregados
Agricultura	U\$ 750.000,00
Comércio Varejista	U\$ 6.500.000,00
Construção	U\$ 31.000.000,00
Agência de viagens	U\$ 3.000.000,00 (em comissões)
Serviços e negócios pessoais.....	U\$ 6.500.000,00
Exceto	
• Arquitetura, engenharia, pesquisas, serviços de mapeamento.....	U\$ 4.500.000,00
• Serviços de limpeza a seco e limpeza de carpetes.....	U\$ 4.500.000,00

Fonte: Small Business Administration

No Brasil, a importância das micro e pequenas empresas é reconhecida inclusive na Constituição Federal, em seu artigo 179:

A União, os Estados, o Distrito Federal e os Municípios dispensarão às microempresas e às empresas de pequeno porte, assim definidas em lei, tratamento jurídico diferenciado, visando a incentivá-las pela simplificação de suas obrigações administrativas, tributárias, previdenciárias e creditícias, ou pela eliminação ou redução destas por meio de lei. (BRASIL, 2004).

Quanto ao critério de classificação das empresas, o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) adota parâmetros que também obedecem a critérios de número de empregados e valor de faturamento bruto anual, relacionando-os com a atividade econômica da empresa (SEBRAE, 2004).

A Tabela 3 apresenta a classificação do SEBRAE em relação ao número de empregados

Tabela 3 – Classificação de empresas quanto ao número de empregados

Porte/ Setor	Indústria	Comércio e Serviços
Microempresas	Até 19	Até 9 empregados
Empresas de Pequeno Porte	De 20 a 99	De 10 a 49
Médias	De 100 a 499	De 50 a 99
Grandes	500 ou mais	100 ou mais

Fonte: SEBRAE (2004)

Em relação ao faturamento bruto anual, SEBRAE (2004) disponibiliza informações que os critérios estão baseados no Estatuto da Pequena Empresa, regulamentado pela Lei Federal nº 9.841/99, atualizada pelo Decreto nº 5.028/04 e BRASIL (2005) disponibiliza informações sobre os valores estabelecidos para micro e pequenas empresas no “Simples Federal” regulamentado pela Lei Federal 9.317 e atualizada pela lei 11.196 de 21 de novembro de 2005.

Na Tabela 4, é apresentada a classificação quanto ao faturamento tanto pelo Estatuto da Pequena Empresa quanto pelo Simples Federal.

Tabela 4 – Classificação de empresas quanto ao faturamento

Porte/ Setor	Estatuto das MPE	Simples Federal
Microempresas	Até R\$ 433.755,14	Até R\$ 240.000,00
Empresas de Pequeno Porte	Acima de R\$ 433.755,14 até R\$ 2.133.222,00	Acima de R\$ 240.000,00 até R\$ 2.400.000,00

Fonte: SEBRAE (2004), SEBRAESP(2006)

Existe, no Brasil, um total de 4,6 milhões de empresas das quais segundo o Instituto Brasileiro de Geografia e Estatística (IBGE) 99% são micro e pequenas empresas (SEBRAE, 2004).

Longenecker, Moore e Petty (1997) registram que o parâmetro mais utilizado é o número de empregados, mas que o melhor critério depende mesmo do propósito do usuário. Assim, para esse trabalho, será adotada a classificação de pequena empresa fornecida pelo SEBRAE.

As pequenas empresas não se diferem das grandes somente por causa de seus critérios de classificação. Uma outra diferença é quanto ao tipo de gestão da pequena empresa. Longenecker, Moore e Petty (1997) informam que embora os gerentes de grandes e pequenas empresas desempenhem funções similares, o trabalho deles é um tanto diferente.

Mendes (2003) afirma que tanto empresas grandes como pequenas exigem um processo gerencial na busca da produtividade e lucratividade, mas Longenecker, Moore e Petty (1997) observam que as pequenas empresas sofrem constantes mudanças em suas necessidades organizacionais e gerenciais, tornando-se extremamente vulneráveis a ter um gerenciamento fraco, enfrentando as dificuldades de pequenas contas bancárias e assessoria gerencial limitada.

4.2 Pequenas empresas e tecnologia de Informação

Conforme a observação de Beraldi, Escrivão Filho e Rodrigues (2000), existem organizações fornecedoras de *software* e hardware voltando-se para os problemas das pequenas empresas e oferecendo serviços personalizados a um menor custo. Dessa forma, possibilitam às pequenas empresas acesso à recursos de tecnologia de informação.

A tentativa de utilização dos recursos de tecnologia de informação pelas pequenas empresas se justifica pela busca da vantagem competitiva. De acordo com Zimmerer e Scarborough (1994), aquelas pequenas empresas que aprendem a explorar o uso da informação e tecnologia para atender melhor seus clientes conseguem obter ganhos de competitividade, assim como as pequenas empresas que coletam e fazem uso da informação disponível para gerenciar suas operações de forma mais eficiente, também conseguem obter vantagem competitiva. A tecnologia está disponível para impulsionar a produtividade, controlar estoques e fluxo de caixa, identificar necessidades e preferências de clientes e antecipar problemas e oportunidades.

De forma mais específica, quanto às pretensões do uso da tecnologia de informação nas pequenas empresas, Vidal (2005) e Moraes (2005, p.111) destacam os benefícios esperados como sendo os seguintes:

- Redução do esforço necessário para gerenciar o negócio;
- Crescimento e expansão do negócio;
- Maior disponibilidade de informação causando melhoria do controle e planejamento;
- Melhoria de atendimento ao cliente;
- Maior possibilidade de delegar autoridade;
- Agilidade na aquisição da informação;
- Evitar retrabalho;
- Acompanhar a modernização

Por meio de descrições apresentadas na literatura com relação à forma de gerenciamento e estratégia na pequena empresa, bem como estudo apresentado por Moraes (2005), pode-se presumir que os responsáveis pelas pequenas empresas ainda não tem um uma ampla visão da estratégia ou da

integração da tecnologia, pessoas e sistema de informação, mas os benefícios esperados com sua utilização de forma geral conduzem às expectativas descritas na literatura para o uso da tecnologia de informação como vantagem competitiva nas empresas de grande porte.

Apesar do fato de Zimmerer e Scarborough (1994) apontarem algumas desvantagens em relação à utilização da tecnologia de informação nas pequenas empresas, destacando o custo elevado em relação a baixa disponibilidade financeira das pequenas empresas; rápida obsolescência de equipamentos e dificuldade de alimentação do sistema com informações corretas, além da relutância dos funcionários por medo da perda de seus empregos e dos riscos para a saúde devido ao uso incorreto do computador por muitas horas, o fato é que o computador está presente no ambiente das pequenas empresas. Há 20 anos atrás, McFarlan (1984) já registrava que a utilização de computadores afetara também as pequenas empresas.

No Brasil, o computador está presente não só nas pequenas, como também nas micro empresas, pois, de acordo com SEBRAE-SP (2003), 47 % das micro empresas possuem computador, índice este que se eleva para 61% quando se trata de indústrias, e chega a 81% quando se trata de empresas de pequeno porte. Embora Moraes (2005, p.16), com base nos mesmos dados de SEBRAE (2003), comente que o índice de 47% nas micro empresas ainda é baixo e que o grau de informatização tende a ser maior nas empresas mais antigas pois estas são em geral de maior porte e melhor estruturadas, é possível afirmar que a informatização é realidade nas pequenas empresas.

Dados apresentados pela Federação das Indústrias do Estado de São Paulo (FIESP) e Faculdade de Economia, Administração e Contabilidade da USP (FEA) (2004, p.9) demonstram que os computadores de mesa (*desktop*) e computadores portáteis (*notebook*), disponíveis nas grandes empresas, estão também presentes nas pequenas empresas, porém em menor quantidade em função do próprio custo do equipamento. Apenas os equipamentos mais específicos como *palmtops*, computadores de grande porte e servidores com tecnologia *risk* ainda não haviam sido adquiridos pelas pequenas empresas na época da pesquisa. A Quadro 9 mostra em detalhes os números de equipamentos (em média) presente nas empresas de diferentes portes.

Porte	Média de equipamentos em 2003					
	Micro Desktop	Note Book	Palm Top	Servidor Intel	Servidor Risc	Computador Grande Porte
Grande	698	81	36	30	5	1
Média	51	4	1	3	0	0
Pequena	11	1	0	1	0	0
Micro	3	0	0	0	0	0

Quadro 9 - Média de computadores por porte de empresa em 2003
Fonte: FIESP/FEA-USP (2004, P.9)

De acordo com SEBRAE (2003), dentre os modelos encontrados nas pequenas e micro empresas do Estado de São Paulo é possível encontrar computadores Pentium 1/2/3 em 53% das empresas, 16% utilizam Pentium 4/AMD/Atlon e 23% fazem uso de equipamentos do tipo 286/386/486/586. O gráfico 2 ilustra estes dados.

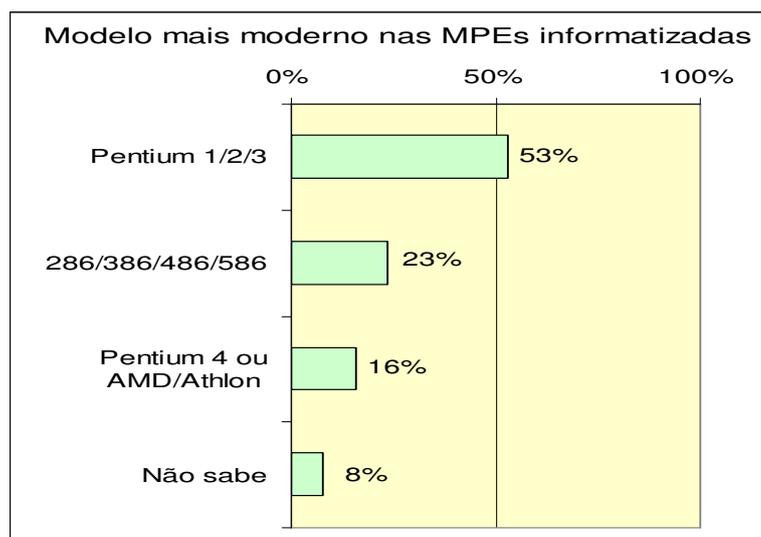


Gráfico 2 - Modelo de computador utilizado nas MPE
Fonte : SEBRAE (2003)

Meireles (2005) realizou um estudo sobre a “Administração de Recursos de Informática” coletando sua amostra entre médias e grandes empresas nacionais de capital privado e apurou que:

- 49% utilizam equipamentos Pentium 4;
- 25% utilizam equipamentos do tipo Pentium 3;
- 16% utilizam equipamentos tipo Pentium 2;

- 7% equipamentos do tipo Pentium 1;
- 2,4% utilizam outros tipos de computadores;
- 0,6% utilizam-se de XT e 486.

Comparando os dados de SEBRAE (2003) e FIESP-FEA/USP (2004) com o estudo realizado por Meireles (2005), pode-se verificar a conformidade de características de equipamentos encontrados nas pequenas empresas com as médias e grandes, apesar de apontar para uma presença bem maior de equipamentos mais modernos (49% já utilizam Pentium 4) nas grandes empresas, fato mais uma vez explicado em função do maior poder aquisitivo e, conseqüentemente, maior investimento na área de tecnologia por parte das empresas de maior porte.

Para que os computadores funcionem é necessário um *software* fundamental chamado de sistema operacional, o qual permite que o computador possa ser utilizado e que outros equipamentos, tais como impressora, *modem*, *scanners*, sejam por ele gerenciados, bem como outros aplicativos, como editores de texto, planilhas eletrônicas, gerenciadores de banco de dados, possam funcionar de forma adequada. Como exemplo de sistema operacional podem ser citados o *Windows*, desenvolvido pela empresa *Microsoft*, e o Linux, desenvolvido sob a filosofia do *Software Livre*.

Dados da FIESP/FEA-USP (2004) mostram que, quando se trata dos computadores que atuam como servidores, o sistema operacional mais utilizado nas pequenas empresas é o *Windows*, com 78%, praticamente empatado com o percentual de utilização do mesmo sistema operacional nas micro (77%), médias (79%) e com a média de utilização nos diferentes portes (79%). Em segundo lugar, encontra-se o Linux, presente em 26% das pequenas empresas, seguido de outros sistemas operacionais. Como reflexo desta maior utilização, verifica-se o *Windows* sendo utilizado com destaque nos diversos tipos de aplicações executadas em servidores. Ainda de acordo com FIESP/FEA-USP (2004), este sistema operacional lidera as funções de servidores quando se trata de rede, banco de dados, aplicativos, correio eletrônico, internet e segurança, apesar do Linux ter presença significativa. A Tabela 5 detalha os tipos de sistemas operacionais utilizados nos servidores nas empresas de diferentes portes, e a Tabela 6 mostra o percentual de

funções executadas por cada sistema operacional dentro do ambiente da pequena empresa.

Ao se tratar de computadores atuando como servidores percebe-se a obrigatoriedade da existência da estrutura de redes de computadores no ambiente da pequena empresa. Segundo FIESP/FEA-USP(2004, p.8), 91% das pequenas empresas possuem uma estrutura de rede de computadores construída. Esta estrutura é responsável por possibilitar a comunicação entre computadores proporcionando facilidades para compartilhamento de informações. Freitas et al. (2004) registram que, no Rio Grande do Sul 54,8% das pequenas empresas possuem todos os seus equipamentos conectados em rede.

Tabela 5 - Sistemas Operacionais utilizados em servidores

	Micro	Pequena	Média	Grande	Total
Windows	77%	78%	79%	88%	79%
Linux	10%	26%	56%	56%	34%
Novell	10%	12%	20%	23%	14%
Unix	0%	2%	13%	59%	9%
Mainframe	0%	0%	2%	12%	2%
Outros	7%	9%	8%	11%	9%

Fonte: Adaptado de FIESP/FEA-USP (2004, p.9)

Tabela 6 - Funções utilizadas nos servidores nas pequenas empresas.

	Rede	Banco de dados	Aplicativos	Correio eletrônico	Servidor internet	Segurança
Windows	68%	52%	31%	45%	45%	26%
Linux	16%	12%	5%	11%	16%	14%
Novell	10%	5%	2%	1%	1%	1%
Unix	1%	1%	1%	0%	0%	1%
Outros	2%	4%	3%	2%	2%	3%

Fonte: Adaptado de FIESP/FEA-USP (2004, p.13)

O *Windows* predomina também quando se trata do sistema operacional utilizado nas estações de trabalho. Segundo dados do SEBRAE (2003) 95% das MPE utilizam *Windows* como sistema operacional. O Linux continua em segundo lugar, a exemplo de sua utilização como servidor, porém agora com uma utilização bastante pequena (2%).

Detalhes sobre a utilização de diferentes sistemas operacionais nas estações de trabalho são apresentados no gráfico 3.

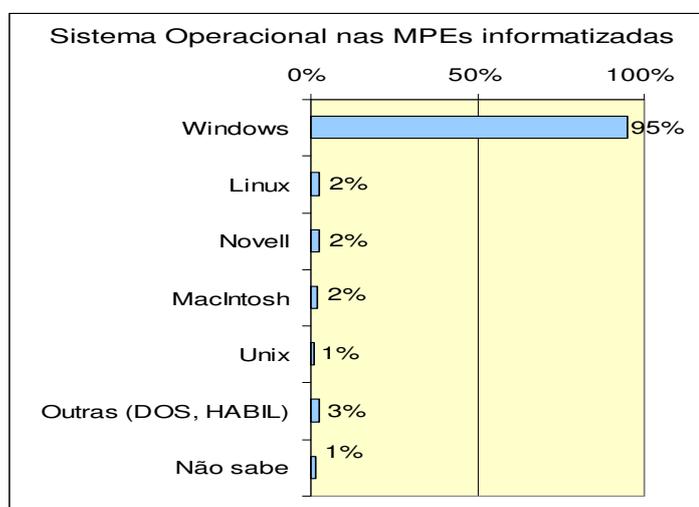


Gráfico 3 - Sistemas operacionais utilizados nas MPE

Fonte: SEBRAE (2003)

Fator de grande importância na existência de um sistema integrado de gestão é a utilização do banco de dados, por isso torna-se relevante apresentar as informações de FIESP/FEA-USP (2004) que apontam para a presença de diferentes gerenciadores de banco de dados nas pequenas empresas, sendo que 78% delas faz uso deste tipo de aplicativo. Os programas relativos a banco de dados encontrados foram: Clipper, Access, Progress, MySQL, SQLServer, Oracle e DB2. Os mais utilizados são Access (18,8%) e Clipper (18,7%). Como informação técnica, vale lembrar que Clipper é o nome dado à uma linguagem de programação do final dos anos 80 que registra os dados em arquivos que obedecem à um padrão chamado xBase.

Tabela 7 - Gerenciadores de Banco de dados utilizados nas pequenas empresas

Clipper	Access	Progress	MySQL	SQLServer	Oracle	DB2	Outros
18,7%	18,8%	1,5%	3,8%	14%	2,6%	0,8%	32,4%

Fonte: Adaptado de FIESP/FEA-USP (2004, p.15)

A Tabela 7 retrata o percentual que cada gerenciador de banco de dados ocupa no ambiente da pequena empresa. Os gerenciadores de banco

de dados dão suporte às operações transacionais de coleta de dados para que os sistemas de informação possam funcionar.

Outro fator importante é a existência de acesso à internet. Reis (2003) considera a distribuição do *software* via internet como uma das características importantes do *Software Livre*. Assim, a empresa que pretende dele fazer uso precisa de alguma forma de conexão à internet, uma vez que todas as atualizações, novas versões e suporte é normalmente disponibilizado para acesso por meio da internet. Mesmo que a empresa contrate o serviço de terceiros para a instalação e implementação de melhorias e correções no *software*, provavelmente este serviço necessitará de algum acesso à *internet*, seja para auxílio remoto à empresa, seja para seu próprio trabalho. Os dados de SEBRAE (2003) apontam para a existência de acesso à *internet* em 54% das MPE, enquanto dados da FIESP/FEA-USP (2004) detalham a forma utilizada para acesso à *internet* na pequena empresa, com destaque para o fato de que mais da metade (55%) das pequenas empresas já não dependem mais de acesso à internet por meio de linha telefônica “discada” e utilizam uma linha de comunicação com velocidade de até 300 Kbps, conforme é ilustrado na Tabela 8. No entanto, Freitas et al. (2004), indicam um percentual de 25% de pequenas empresas utilizando esta forma de acesso no Rio Grande do Sul, enquanto os 75% restantes ainda utilizam a linha telefônica. A Tabela 8 apresenta estas informações.

Tabela 8 – Forma de acesso à internet nas pequenas empresas.

Provedor Grátis	Provedor Pago	Linha Comunicação Até 300 Kbps	Linha Comunicação Até 600 Kbps	Linha Comunicação Acima de 600 Kbps
7%	26%	55%	16%	5%

Fonte: Adaptado de FIESP/FEA-USP (2004)

As pequenas empresas, de forma geral, possuem uma estrutura de tecnologia de informação, contudo ainda há muito espaço para crescimento e

aperfeiçoamento neste segmento. A próxima seção mostra algumas implicações do uso do sistema de informação na pequena empresa.

4.3 Pequenas empresas e *softwares* de gestão

Conforme Bigaton (2005), as empresas de pequeno porte têm carência de instrumentos de gestão da informação que possam auxiliá-las a aumentar a produtividade e capacidade de concorrência em ambientes competitivos.

De acordo com Liczbinski (2002, p. 53), o Sistema de Informação Gerencial é provavelmente o tipo mais utilizado nas pequenas empresas.

Os sistemas de informação representam um elemento de apoio à gestão porque podem fornecer aos gestores informações que objetivam esclarecer fatos e condições para tomada de decisão de forma consciente sobre o que os fatos representam e quais suas conseqüências (LICZBINSKI, 2002).

Bigaton (2005) lembra que mesmo que nem sempre os responsáveis pela administração nas pequenas empresas possuam consciência das diferenças entre os níveis organizacionais (estratégico, tático, operacional), uma vez que muitas vezes acumulam funções nos diferentes níveis, é possível a utilização do sistema de informação para auxiliá-los.

As pequenas empresas pouco se utilizam da adoção de um pacote integrado de sistemas de um único fornecedor, preferindo desenvolver internamente seus próprios aplicativos, adquirir aplicativos de terceiros ou realizar as tarefas manualmente (FIESP/FEA-USP, 2004; BIGATON, 2005; FREITAS et al.,2004).

Freitas et al. 2004 sugerem a baixa utilização de aplicativos mais elaborados e complexos como conseqüência do alto custo de aquisição e manutenção destes para a pequena empresa, mas destaca utilização de aplicativos principalmente nas áreas contábil-financeira, materiais, compras e vendas.

Informações da FIESP/FEA-USP (2004) também indicam a utilização de *software* para auxiliar diferentes áreas na pequena empresa, concordando com uma baixa utilização de pacotes ERP, além da utilização de *softwares* desenvolvidos pelas próprias empresas, ou utilização de pacotes do tipo Office

(editores de texto e planilhas de cálculo) para apoiar as atividades da pequena empresa. Um perfil das áreas de utilização de aplicativos é descrito a seguir:

- **Área comercial:** somente 10% das pequenas empresas utilizam pacotes de ERP para apoio à esta função, auxiliando principalmente as tarefas de cadastro de clientes, pedidos, análise de crédito, emissão de nota fiscal, controle de estoque, relatório de vendas e controle de cargas e fretes.
- **Área de produção:** nesta área a pequena empresa utiliza *software* para apoio às atividades de estoque, produto, produção, ordem, custo, controle de qualidade, manutenção e projeto. A utilização de pacotes integrados nesta área é feita por 15% das pequenas empresas.
- **Área de suprimentos:** também utiliza ERP em 15% das pequenas empresas, sendo que as atividades informatizadas nesta área objetivam melhorar as atividades de apoio às compras, reduzir custos, reduzir falta de peças e material de reposição, reduzir prazo de atendimento a clientes, aumentar o poder de negociação com fornecedores.
- **Área administrativa:** embora não sejam apresentados percentuais sobre a utilização dos sistemas integrados de gestão, os aplicativos são utilizados nas atividades de contas a pagar, contas receber, contabilidade, livros fiscais, além de uma baixa utilização na gestão de recursos humanos, orçamento empresarial e ativo fixo. Existe também o uso de aplicativos do tipo Office (planilhas de cálculo e editores de texto).

Além das aplicações em áreas específicas, FIESP/FEA-USP (2004) apresentou dados da presença de outras categorias de sistemas de informação, conforme ilustrado no Gráfico 4, em um percentual considerável das pequenas empresas. Os sistemas encontrados foram: Business Intelligence e Executive Information System (BI/EIS), CRM (Customer Relationship Management), SCM (Supply Chain Management), WFM (Work Flow Management) e CAD (Computer Aided Design).

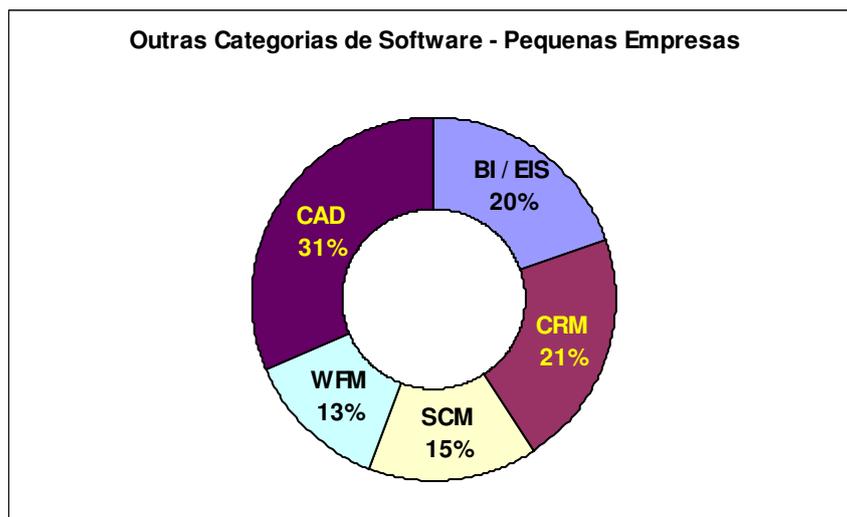


Gráfico 4 - Outras categorias de *software* encontradas nas PE

Fonte: Adaptado de FIESP/FEA-USP (2004, p.15)

Bigaton (2005) escreve que o sistema de informação gerencial poderia estar mais difundido nas pequenas empresas, porém isto esbarra no problema do custo relacionado à situação financeira das empresas, uma vez que os sistemas existentes no mercado, considerados “prontos”, quase sempre ultrapassam os valores disponíveis nas pequenas empresas para investimento nestes recursos. Bigaton (2005) destaca ainda que as empresas que desenvolvem estes sistemas não estão dispostas a considerar um “pequeno retorno financeiro” para o desenvolvimento de um sistema específico, e, desta forma o benefício da utilização de um sistema de informação ainda não está totalmente difundido nas pequenas empresas.

Sob esta característica, os aplicativos *Open Source* levam vantagem pelo baixo custo de aquisição, não sendo necessário o pagamento de licença de uso ou valor cobrado pelo desenvolvimento do sistema, mas existe um detalhe que precisa ser considerado: a questão de pessoal capacitado à instalação, parametrização e possíveis alterações necessárias.

4.4 Pequenas empresas e recursos humanos na área de informática

Para Reis e Escrivão Filho (2005), a literatura sobre Recursos Humanos tem se dedicado a tratar de conceitos globais aplicados à grandes empresas, mas quando se trata das práticas de Recursos Humanos ou treinamento

aplicados à pequena empresa não só a literatura mas, conseqüentemente, o conhecimento se torna escasso. Os poucos estudos realizados indicam uma estrutura sem autonomia e bastante simples de gerenciamento, utilizando técnicas tradicionais.

A forma como as pequenas empresas gerenciam o fator humano não diminui a relevância nem tampouco a necessidade das empresas contarem cada vez mais com trabalhadores qualificados para acompanhar o ritmo das mudanças (REIS e ESCRIVÃO FILHO, 2005).

Hehn (1999, p.17), ao tratar da implementação de sistemas ERP, descreve a grande importância do fator humano neste processo e a necessidade da interação das pessoas com seus valores e crenças às políticas e ao sistema de recursos humanos da empresa. O autor considera o recrutamento, seleção, avaliação de desempenho, remunerações, premiações, plano de carreira, papéis e responsabilidades dos funcionários, além dos objetivos e a estrutura organizacional, como um conjunto chamado "*peopleware*".

Sobre este conceito (*peopleware*), Hehn (1999) quer mostrar que empresas são compostas por pessoas com inseguranças, sentimentos, fantasias e outros aspectos que precisam ser considerados e adequadamente tratados para que um projeto (não somente o de ERP) seja realizado com sucesso.

Particularmente em um projeto de implementação de ERP, considerar as pessoas envolvidas é fator de vital importância e oferece um grande campo de estudo e abrangência atingindo questões de cultura organizacional, alinhamento estratégico e gestão do conhecimento, entre outras, que fogem ao foco da preocupação deste trabalho. Porém, a necessidade de pessoas capacitadas no momento da instalação dos aplicativos é importante para a verificação da disponibilidade de mão-de-obra capacitada para a realização de tal tarefa no ambiente da pequena empresa.

De acordo com FIESP (2004), um percentual de 56% das pequenas empresas possuem um departamento ou uma área criada para tratar de TI dentro das empresas, com uma média de dois funcionários neste departamento e uma pessoa contratada como prestador de serviços (terceiro).

O Gráfico 5 demonstra o número de pessoas utilizadas na área de tecnologia de informação nas micro, pequenas, médias e grandes empresas, além de mostrar uma média geral do número de pessoas (total). Como destaque aparece a grande diminuição de empresas que utilizam pessoal próprio em TI nas pequenas (56% utilizam) e micro-empresas (0%). No Gráfico 6, verifica-se o número médio de pessoas destinadas à área de TI na pequena empresa.

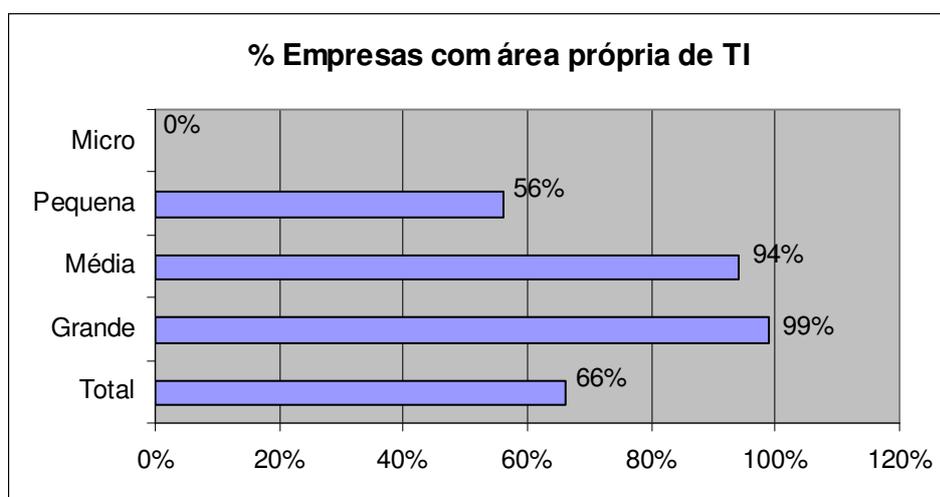


Gráfico 5 - Empresas com área de própria de TI

Fonte: Adaptado de FIESP/FEA-USP (2004, p.15)

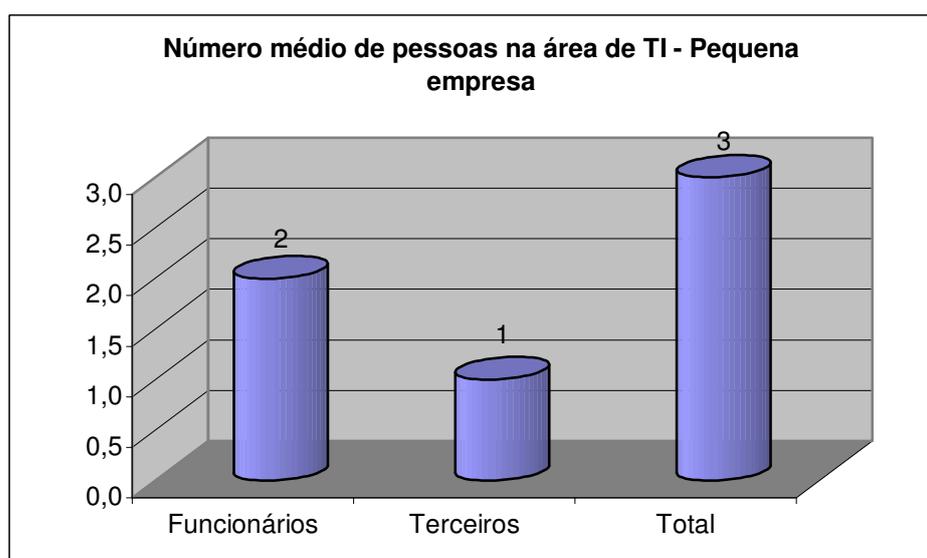


Gráfico 6 - Número de pessoas na área de TI

Fonte: Adaptado de FIESP/FEA-USP (2004, p.15)

Os dados apresentados por Freiras et al. (2004), originados de um estudo realizado no estado do Rio Grande do Sul, registram um número bastante aproximado de pessoas pertencentes às atividades de TI naquele estado (2,9 pessoas), com uma taxa de terceirização de atividades de 5,06%.

Vidal (2005), Moraes (2005) e Freitas et al.(2004) também comentam que as pequenas empresas se utilizam de serviços contratados para o desenvolvimento de sistemas de informação, confirmando a presença de terceiros na área de TI.

Quanto aos usuários dos sistemas de computador nas pequenas e médias empresas, Vidal (2005) escreve que estes não têm experiência ou conhecimento sobre os equipamentos e sistemas, fato que contribui para dificultar a seleção, aquisição e implantação dos aplicativos. Esta mesma perspectiva é reforçada por Moraes (2005, p.112), que apresentou um estudo de caso realizado em quatro empresas, sendo que apenas uma tinha alguém com formação na área de informática, enquanto o restante dos funcionários que tinham contato com o sistema de informação, inclusive os dirigentes, estavam restritos ao funcionamento do aplicativo que utilizavam, embora tenham feito cursos em escolas de informática. Freitas et al. (2004) também comentam que além da existência de um menor grau de especialização nos profissionais responsáveis pela área de TI nas pequenas empresas, é comum que este profissional acumule cargos, não tendo dedicação exclusiva à área de TI.

4.5 Considerações

Analisando todos os fatores envolvidos na adoção de um sistema integrado de gestão, percebe-se que não é um processo simples, que atinge apenas uma área da organização. Ele envolve questões de processos, tecnologia, pessoas e as conseqüentes mudanças organizacionais. Além disso, as pequenas empresas têm recursos monetários escassos, fazendo que não sejam cogitados projetos milionários com prazo de implantação de dois anos, fato que é comum em projetos de implantação de sistema de informação em grandes empresas. (MENDES, 2003).

Com base nas informações apresentadas neste capítulo, pode-se concluir que existe um ambiente de informatização nas pequenas empresas uma vez que estas já utilizam aplicativos de computador para auxiliá-las no gerenciamento, bem como possuem uma estrutura de equipamentos e redes que suportam estes aplicativos. Embora a presença dos sistemas integrados seja pequena, à medida que sua divulgação cresce, a sua utilização tende a crescer a exemplo e impulsionada pelo uso que já acontece nas grandes empresas.

Freitas et al. (2004) comentam a espera de um maior crescimento da informatização das pequenas empresas nos próximos anos do que nas grandes empresas, uma vez que estas já possuem um maior nível de informatização, enquanto as pequenas ainda têm espaço para a informatização de diversas atividades.

Outro fator a ser considerado é que já existe a presença de pessoas específicas nas pequenas empresas para a área de TI (56% tem área própria de TI), e mesmo que estas pessoas não sejam altamente qualificadas, conforme informa a literatura, a falta de experiência dificulta o processo de implantação do sistema de informação, mas não o impede. O que não se pode afirmar é que o número de pessoas na área de TI é ideal (provavelmente não é) ou que seja suficiente para desempenhar o papel da área em sua plenitude. De qualquer forma, o fato de ter um departamento de TI facilita a adoção de um *software* livre, pois subentende-se que existam pessoas para trabalhar em sua parametrização e adaptação às necessidades específicas da empresa, o que é reforçado pelo fato de que a pequena empresa parece ter uma tendência ao desenvolvimento próprio de sistemas.

Nota-se também indicadores de que a pequena empresa adota serviços prestados por terceiros para suprir as necessidades da área de informática. Seria necessário avaliar uma possível não aceitação dos desenvolvedores terceirizados em realizar os serviços de manutenção e codificação em relação ao *software* livre, uma vez que estes provavelmente já têm seus próprios produtos, o que levaria à necessidade de formar pessoal especializado na própria empresa ou torna a contratação de mão-de-obra terceirizada bastante difícil. Entretanto alguns grupos que desenvolvem *software* livre oferecem

serviços de consultoria para implantação dos sistemas, cobrando apenas pelos serviços e não pelo *software*.

Assim, apesar da falta de informações sobre que tipo de atividade o pessoal contratado exerce, seja na manutenção de equipamentos, estrutura de rede ou desenvolvimento de sistemas, fica o registro de que a pequena empresa pode recorrer a um profissional terceirizado para suprir suas necessidades como uma situação normal dentro de suas características na área de informática.

Barbieri (1999, p.131) usa o termo “transferência de tecnologia” para indicar o processo onde a empresa passa a dominar o conjunto de conhecimentos que não foi por ela produzido e afirma que para que esta transferência realmente ocorra é necessário que a empresa utilizadora da tecnologia também se esforce para desenvolver tecnologia própria. De forma análoga, saber operar um *software* não significa que se tenha dominado a tecnologia que ele disponibiliza. Visto dessa forma, o desenvolvimento interno do aplicativo, com a absorção da tecnologia por ele utilizada poderia proporcionar o desenvolvimento da tecnologia própria, gerando um produto realmente integrado às características da pequena empresa.

Esta visão não é a única possível, pois o mesmo Barbieri (1999, p.132) lembra que adquirir a tecnologia externa também pode eliminar etapas do processo de inovação e provocar ganhos de tempo e de recursos com avanços em relação à geração e organização de conhecimentos.

Longenecker, Moore e Petty (1997) alertam para o fato de que a existência de uma série de vantagens e desvantagens associadas à implantação de um SIGE nas pequenas empresas leva à uma análise, necessária, tanto dos benefícios quanto dos riscos. Esta análise é perfeitamente aplicável e necessária à opção das pequenas empresas pelo uso do *software* livre.

Este capítulo apresentou características das pequenas empresas com relação à área de Tecnologia da Informação, seus equipamentos, uso de aplicativos e recursos humanos. Isto é importante no contexto deste trabalho para que se possa compreender melhor a possibilidade de instalação dos aplicativos de *software* livre selecionados para a pesquisa no ambiente da

pequena empresa, relacionando estas informações ao questionário (tratado com mais detalhes na seção 3 do capítulo 5) aplicado na fase de testes de instalação. As principais questões envolvidas são referentes ao tipo de *hardware* necessário para instalação do aplicativo, *softwares* adicionais necessários para que o aplicativo possa funcionar corretamente e dificuldades de instalação.

O próximo capítulo tratará da metodologia utilizada neste trabalho.

Capítulo 5 – Procedimentos metodológicos

Esse capítulo descreve os procedimentos metodológicos adotados para o estudo, apresentando o tipo de pesquisa, coleta de dados e limitações da mesma. Com isto, espera-se apresentar informações condizentes com a definição de projeto de pesquisa adotada por Yin (2001), em que se deve apresentar uma seqüência lógica em que os dados se conectam às questões de pesquisa iniciais do estudo. Yin (2001) conclui ainda que em última análise, essa seqüência lógica se conecta com as conclusões, passando por pelo menos quatro problemas: quais questões estudar, quais dados são relevantes, quais dados coletar e como analisar os resultados.

5.1 Metodologia

Esse estudo foi classificado como tendo um referencial teórico qualitativo-positivista. Na definição de Martins (1995), o positivismo não aceita uma realidade diferente daquela que não seja a realidade dos fatos que possam ser observados. Esta é uma das idéias básicas do positivismo: a busca da explicação dos fenômenos se dá através da evidência dos fatos, por meio de sua observação e de seus relacionamentos.

5.1.2 Método

Nessa seção, serão descritas as questões metodológicas quanto ao objetivo, delineamento e instrumentos de coleta.

- **Objetivo:** Dane (1990) explica que o objetivo de uma pesquisa descritiva é capturar as características de um objeto, uma pessoa ou um evento em um determinado período no tempo através da coleta de

dados, mas estes dados podem mudar com o decorrer do tempo. Assim, o objetivo metodológico desse trabalho foi classificado como descritivo, pelo fato de procurar evidenciar quais são as características presentes nos *softwares* de gestão construídos sob o modelo de *software* livre aberto e a viabilidade de sua instalação nas pequenas indústrias.

Ainda de acordo com Dane (1990), a pesquisa descritiva envolve examinar um fenômeno para poder defini-lo de forma mais abrangente, ou diferenciá-lo de um outro fenômeno.

- **Delineamento:** em concordância com o objetivo, foi escolhido como delineamento do estudo o “Levantamento” que, na opinião de Gil (1998), é adequado para estudos descritivos, sendo indicado por exemplo para estudos do tipo censo, onde a identificação de características é o que se deseja conhecer.
- **Instrumentos de coleta:** como instrumentos de coleta foi utilizado um questionário utilizando questões fechadas e abertas, obtendo-se as respostas por meio da realização da instalação dos *softwares* selecionados.

5.2 As variáveis de pesquisa

Kalpik e Fertalj (2004, p.196) explicam que ERP são complexos, então sua avaliação também é uma tarefa muito complexa, além do mais, não existe um método de avaliação que seja totalmente aceito.

Alguns métodos amplamente aceitos retratam a avaliação do desenvolvimento do *software* em relação à organização ou concentram-se na escolha de fornecedores de serviços. Outros objetivam analisar o mercado. (KALPIC e FERTALJ, 2004).

Outra dificuldade se encontra no grande número de aplicativos disponíveis no mercado. De acordo com Best *Software* (2004), duas maneiras são utilizadas na tarefa de avaliação:

- Avaliação interna: quando a empresa resolve fazer a avaliação do *software* com recursos próprios. O que normalmente ocorre é a nomeação de um funcionário ou uma equipe de funcionários que realizam o levantamento de uma lista de requisitos e funcionalidades baseada no que se espera que o *software* tenha para atender as necessidades da empresa e a partir disto realizam a avaliação.
- Consultoria de *software*: ocorre quando a empresa contrata consultores para gerenciar o processo de avaliação. Neste caso pode-se contar com a experiência dos consultores contratados, bem como relatórios detalhamento dos produtos e serviços a serem adquiridos.

Com base no objetivo estabelecido para esta pesquisa e de forma análoga ao trabalho de Kalpic e Fertalj (2004), que definiu suas próprias necessidades, e no método de avaliação interna que funciona de forma similar, foram estabelecidos três grupos de variáveis de pesquisa: requisitos de instalação, funcionalidades e usabilidade. Além destas variáveis, foi definida como variável dependente o próprio ambiente da pequena empresa em relação à situação de tecnologia, pessoas e uso dos sistemas de informação. Esta variável dependente é apresentada com mais detalhes na seção 5.6 e está diretamente relacionada às informações apresentadas no capítulo quatro e na parte do questionário que identifica dados do sistema analisado em relação às características da pequena empresa. Cada uma das outras variáveis será apresentada em detalhes a seguir:

5.2.1 Requisitos de instalação

O objetivo desta variável é identificar quais são os requisitos de hardware e *software* adicional necessários para que o *software* analisado possa ser completamente instalado e utilizado. Em alguns casos, principalmente quando o *software* é desenvolvido para utilização em ambiente *web*, torna-se necessária a instalação de outros programas como, por exemplo, um gerenciador de banco de dados ou servidor de páginas *web*. Para outros casos, basta somente a instalação do próprio *software*. Assim, as respostas que se procuram com relação a essa variável são:

- **Recursos de hardware:**
 - Quais são os recursos de hardware necessários para que o sistema seja instalado? Esses recursos são encontrados no ambiente de tecnologia das pequenas empresas?

- **Software adicional:**
 - Existe a necessidade de instalação de *software* adicional para que o aplicativo possa funcionar adequadamente? Se existe, quais são eles?

- **Recursos de instalação:**
 - O *software* possui um programa instalador que auxilie o processo de instalação?
 - Existe documentação sobre os requisitos e forma de instalação do *software*?

5.2.2 Funcionalidades

Para que o *software* possa atender as necessidades a que ele se propõe, serão analisadas funcionalidades com relação a:

- **Integração com banco de dados:**
 - O *software* é capaz de obter dados de mais de um tipo de banco de dados ou sistema gerenciador de banco de dados? (Isso se torna particularmente importante, caso a empresa já possua dados cadastrados e precise aproveitá-los).
 - É possível criar ou alterar campos e tabelas no banco de dados padrão do sistema para adaptá-lo às necessidades da empresa?
 - Pode-se alterar as telas do *software* com relação aos campos de dados ?
 - O *software* é capaz de gerenciar dados de empresas diferentes (por exemplo: matriz e filiais)?

- **Relatórios e gráficos**
 - Quais são os relatórios oferecidos pelo *software*, de acordo com sua instalação padrão?
 - O *software* possui ferramentas que permitam criar novos relatórios ou alterar relatórios existentes de acordo com a necessidade da empresa?
 - Quais são os formatos sob os quais os relatórios são apresentados? Somente na tela, apenas na impressora, é possível gravar os relatórios em arquivos? Em que formato de arquivo os relatórios podem ser gravados?
 - É possível gerar gráficos com as informações? Que tipos de gráficos?

- **Recursos de integração com outros *softwares***
 - O *software* possibilita comunicação com outros *softwares*, por exemplo, os *softwares* chamados de “pacotes de escritório” (do tipo Office: Word, Excell)?
 - Permite integração com *software* de *e-mail* ou algum tipo programa de envio de mensagens?

- **Segurança**
 - Quais são os recursos de segurança oferecidos pelo *software*?
 - Existe senha para acesso ao *software*?
 - Existem níveis de acesso diferentes, com acesso restrito a diferentes áreas?
 - A que tipo de seguranças as informações gerenciadas pelo *softwares* podem ser impostas (criptografia, segurança oferecida pelo gerenciador de banco de dados, segurança oferecidas pelo sistema operacional)?

- De que forma é estabelecido o sistema de *backup* dos dados? O próprio *software* gerencia o *backup* ou é necessário uma rotina externa ao programa?

5.2.3 Usabilidade

A análise desta variável procura identificar o relacionamento homem-máquina (usuário – sistema) que o *software* pode proporcionar. Este aspecto do *software* faz parte de uma área chamada de usabilidade. A usabilidade é um assunto bastante extenso, chegando a ser tratada como “engenharia de usabilidade” e ocupando destaque dentro de um aspecto maior: a qualidade de *software*.

A definição de usabilidade possui diferentes aspectos e abordagens, mas, para este trabalho, adotou-se a definição de Nielsen (1993) onde devem ser incluídos ao menos os seguintes atributos: facilidade de aprendizado, eficiência de uso, retenção, mínimo de erros e satisfação.

As formas de avaliar a usabilidade também são variadas, e segundo Barcelos (2001), podem ser classificadas em prospectivas, preditivas e objetivas. As técnicas de avaliação ditas prospectivas estão baseadas na opinião do usuário, as preditivas baseadas em modelos ou conhecimento e as objetivas tem sua base em ensaios de interação do usuário ou em sistemas de monitoramento (por exemplo, observação com câmeras).

Para este trabalho, a proposta é a adoção de uma técnica chamada de “Avaliação Heurística”, que é uma técnica preditiva baseada em conhecimento. Esta técnica foi escolhida porque, segundo a descrição de Barcelos (2001), é uma técnica relativamente fácil (pode ser assimilada em meio dia de estudos), rápida (consome um dia para a maioria das avaliações), barata e menos intimidante.

O método de avaliação Heurística, na descrição de Nielsen (1993), baseia-se na observação dos seguintes princípios:

- **diálogo simples e natural:** é desejável que a interface com o usuário seja tão simples quanto possível;

- **falar a língua do usuário:** diálogos não devem ser orientados aos sistemas, mas expressos claramente em palavras, expressões com as quais o usuário esteja habituado;
- **minimizar a carga de memória do usuário:** é importante tornar fácil a visibilidade de objetos, ações e opções;
- **consistência:** não devem existir dúvidas quanto ao significado de diferentes ações, palavras ou situações;
- **retroalimentação:** deve existir, por parte do sistema, informação constante ao usuário sobre o que está sendo feito, e como sua entrada foi interpretada;
- **saídas claramente indicadas:** o sistema deve fornecer condições de saídas ao usuário que facilitem seu controle sobre o sistema, por exemplo, a opção de desfazer determinada ação;
- **atalhos:** de forma a acelerar a interação do sistema para usuários mais experientes;
- **boas mensagens de erro:** as mensagens de erros precisam ser escritas em linguagem clara e evitar o uso de códigos obscuros;
- **prevenção de erros:** sistema deve tentar identificar os pontos de erros mais prováveis e procurar minimizar essas situações;
- **ajuda e documentação:** as informações devem ser fáceis de procurar, enfocadas nas tarefas, listadas em passos concretos e não ser muito extensa.

De forma geral, o objetivo desta variável é responder questões como:

- **Interface com o usuário:**
 - As telas do *software* possuem um visual agradável, de fácil “navegação” e compreensão para o usuário?
 - As informações que aparecem nas telas estão bem posicionadas e possuem tamanho de fonte suficiente à sua compreensão?
 - O *software* apresenta suporte a várias línguas diferentes?
- **Documentação**
 - O *software* possui uma documentação que detalhe sua forma de instalação e funcionamento?

- De que forma essa documentação pode ser acessada? Diretamente nas telas de operações do *software*, no momento em que a dúvida surgir? Em formato de manuais?
- É possível encontrar alguma forma de suporte à instalação, treinamento ou problemas com o *software*? De que forma?

5.3 O Questionário

Uma vez determinadas as variáveis, novamente seguiu-se o exemplo do método de Avaliação Interna e do que foi feito por Kalpic e Fertalj (2004) que desenvolveram um questionário específico com o objetivo de comparar diferentes *softwares* de ERP, uma vez que os métodos existentes não satisfaziam suas necessidades. Para este trabalho foi elaborado um questionário próprio, pois o propósito era a verificação das características do *software* e a possibilidade de instalação do aplicativo no ambiente da pequena empresa, fatores que diferem da proposta dos métodos existentes.

O apêndice A, apresenta o questionário utilizado para o levantamento dos dados da pesquisa. Nesta seção, pretende-se explicar quais são as informações e os objetivos das questões que compõem o questionário.

5.3.1 Parte 1

A primeira parte do questionário se destina a identificar o *software* analisando:

- **Nome** : como o *software* é conhecido ou divulgado;
- **Desenvolvedor**: responsável pelo desenvolvimento (grupo, empresa, indivíduo);
- **Web site**: endereço onde o *software* pode ser encontrado na *web*,
- **Início de Desenvolvimento**: para determinar desde quanto o *software* está em desenvolvimento;
- **Sistema operacional**: identificar o sistema (ou sistemas) sob os quais é possível executar o *software*;

- **Tipo de licença:** com o objetivo de identificar o tipo de licença adotado, dentre as possíveis licenças de *software*,
- **Classificação:** objetivo de identificar se é um *software* mais adaptado à área de serviços, comércio ou indústria (a identificação se dá através de informação divulgada pelo próprio desenvolvedor ou pela identificação das funcionalidades presentes);
- **Principais usuários:** quem são os possíveis principais usuários em potencial: pequenas, micro ou quaisquer tipo de empresa.

Após a identificação do *software*, o questionário terá uma parte dedicada à primeira variável de pesquisa: viabilidade de instalação. As questões trabalhadas foram divididas em 3 itens: Identificação do hardware, recursos de instalação e *software* adicional.

- **Identificação do hardware:** o objetivo é identificar quais são os recursos mínimos de hardware para que o *software* possa funcionar corretamente. Estas informações ajudam a identificar se é possível encontrar o hardware requisitado pelo *software* no ambiente das pequenas empresas.
- **Recursos de instalação:** o propósito é identificar se durante o processo de instalação, o *software* oferece recursos que auxiliem nesta tarefa. Pretende-se descobrir se existe um instalador ou se o *software* é instalado através de outros meios como, por exemplo, a simples cópia dos arquivos para o equipamento onde será executado. Uma outra preocupação é identificar a forma (se é que ela existe) de proceder a remoção do *software* depois que o mesmo se encontra instalado. Essas informações são úteis para a verificação do nível de conhecimento em informática necessário para que o *software* seja instalado no ambiente das pequenas empresas, uma vez que, segundo a literatura, as pequenas empresas não dispõem facilmente de mão-de-obra qualificada para tais tarefas.
- **Software Adicional:** segundo a própria descrição da literatura, os *softwares* de gestão empresarial estão alicerçados sob uma base centralizada de dados. Este fato impõe pelo menos a existência de mais um *software* funcionando: o servidor de banco de dados. Esta

seção do questionário pretende identificar se a própria instalação efetua a instalação deste recurso, ou ele deve ser instalado de forma individual. Além deste, outro recurso que pode ser necessário, caso o modelo de desenvolvimento tenha optado pelas funcionalidades de um ambiente *web* é a presença de um *software* conhecido como “servidor de páginas *web*”. Para esta questão pretende-se identificar também o nível de dificuldade desta instalação.

5.3.2 Parte 2

A segunda parte do questionário trata da segunda variável de pesquisa: as funcionalidades presentes no *software*.

- **Banco de dados:** as primeiras funcionalidades analisadas dizem respeito ao banco de dados procurando identificar qual é o Gerenciador de Banco de Dados (SGBD) com o qual o *software* se relaciona de forma mais direta (nativa) e, em seguida, se é possível que o *software* utilize outros bancos de dados para armazenagem. Isto é importante pelo fato de que é possível que o SIGE seja desenvolvido sob os moldes do *software* de código aberto, mas utilize como base de dados um *software* cujo custo possa ser inviável para uma pequena empresa. Ainda com relação ao banco de dados procura-se identificar se é possível importar dados de outras bases. Este recurso torna-se interessante quando as pequenas empresas já tiverem dados armazenados e tornar-se necessário importar estes dados para a base sob a qual o SIGE vai trabalhar. Outra questão diz respeito à possibilidade de adaptação e criação de outros campos nas tabelas do banco de dados para adaptar o sistema às necessidades da empresa, e ainda se o sistema permite gerenciar mais de uma empresa, por exemplo: matriz e filial.
- **Gráficos e Relatórios:** o objetivo desta seção é identificar a forma como a informação chega ao usuário, quais são os relatórios presentes no *software*, se existe possibilidade de criação de relatórios necessários ao perfil da empresa e se as informações podem ser apresentadas em forma de gráficos.

- **Recursos de Integração com outros softwares:** o propósito é registrar se o *software* pode integrar-se a outros programas que já estão ou podem vir a fazer parte do ambiente da empresa como, por exemplo, editores de texto e planilhas eletrônicas.
- **Módulos:** relacionar quais são os módulos que compõe o sistema.
- **Recursos de comunicação presentes no software:** as questões desta parte do questionário destinam-se a identificar se é possível de dentro do ambiente do *software* utilizar ferramentas de comunicação como, por exemplo, um recurso de *e-mail*, ou outro tipo de mensagem através da rede de computadores.
- **Segurança:** ainda dentro da identificação de funcionalidades, esta seção do questionário preocupa-se em mapear os recursos de segurança aos dados e à quem é de direito a utilização da informação oferecidos pelo *software*, bem como recursos de cópia de segurança dos dados.

5.3.3 Parte 3

A terceira parte do questionário é referente à terceira variável de pesquisa: usabilidade. Nesta parte a preocupação foi identificar se, uma vez instalado o *software*, é de fácil utilização e possui recursos que possibilitem uma boa interação entre o usuário e *software*.

Baseado na Engenharia de Usabilidade, e utilizando as descrições de Nielsen (1993), foram construídas questões para responder se o *software* traz um ambiente agradável de trabalho, fornecendo informações sobre o seu funcionamento, bem como se existe documentação sobre o *software* e sua operacionalidade.

5.4 Seleção de amostras

Para este estudo foi escolhido como base de dados o repositório de projetos *SourceForge* não só pelo fato de ser o pioneiro em fornecer o serviço de hospedagem de projetos, conforme declara Reis (2003), mas também pelo volume considerável de projetos ali hospedados que chegavam a mais de 100.000 no mês de maio de 2005, o que contribui para uma maior representatividade da amostra (*SOURCEFORGE*, 2005).

Além destes fatores, o *SourceForge* já foi utilizado como base para outros trabalhos envolvendo projetos de *software* livre, por exemplo, Krishnamurthy (2002) utilizou-o como fonte para um artigo intitulado *Cave or community ? An empirical examination of 100 mature Open Source Projects* em que se identificou o número médio de desenvolvedores envolvidos por projeto, volume de mensagens em listas de discussão e correlação entre idade do projeto e número de desenvolvedores. Reis (2003) utilizou os repositórios *SourceForge* e *Fresmeat* como local para identificar *softwares* para caracterização de desenvolvimentos de projeto de *software* livre no que se refere ao processo de desenvolvimento de *software*.

A forma utilizada para a obtenção de amostras foi classificada como não probabilista de acordo com a definição de Marconi e Lakatos (1999, p.53), que define como característica principal da amostragem não-probabilística o fato de não fazer uso de formas aleatórias de seleção. A amostragem não-probabilista também impossibilita certos tipos de tratamento estatístico como, por exemplo, os de erro de amostragem.

A técnica empregada para se obter a amostra foi a de “fases múltiplas, multifásica ou em várias etapas” que, segundo Marconi e Lakatos (1999), consiste na obtenção de uma amostragem mais ampla que é submetida a uma investigação rápida e pouco profunda através da qual se obtém conhecimento para extrair uma amostra menor que será objeto de uma pesquisa mais aprofundada.

No caso deste trabalho, a determinação da amostra inicial ou mais ampla foi feita com o auxílio do próprio mecanismo de busca oferecido pelo repositório *SourceForge.net* utilizando a “busca orientada”. Blattmann e Tristão (2005) definem como “busca orientada” aquelas em que o mecanismo de

busca é utilizado para recuperar informação com base no resultado do uso da lógica *booleana*. De forma prática, significa oferecer ao mecanismo informações que servirão de critério para a validação do resultado como verdadeiro.

Blattmann e Tristão (2005) explicam que os mecanismos de busca enquanto ferramentas disponíveis em ambiente *web*, apresentam características diferentes, possuindo cada um deles suas vantagens e limitações, ficando a critério do pesquisador a utilização destes mecanismos geralmente implícita na qualidade informacional obtida.

Para este trabalho foram realizados testes com o mecanismo de busca para a ambientação e melhor adequação das possibilidades de utilização. Constatou-se a existência do aspecto da informação “categoria do *software*” que deveria ser informada no momento do cadastro do *software* no repositório, mas não foram encontradas informações que pudessem informar a partir de qual data uma categoria foi criada nem se em algum momento houve um recadastramento dos projetos para adequação à novas categorias criadas, nem de que forma as categorias foram estabelecidas, o que leva à percepção de que é o próprio usuário que decide em que categoria enquadrar seu projeto, embora existisse a categoria “ERP” disponível para escolha. Em função deste fato, julgou-se que apenas uma busca orientada por categorias não seria suficiente.

As buscas da fase de teste tornaram possível perceber que os resultados apresentados pelo mecanismo mostravam uma descrição das características dos projetos, o que está de acordo com a descrição dos padrões de resultados de mecanismos de busca apresentada por Blattmann e Tristão (2005). Este fato motivou a escolha da palavra “ERP” para a busca orientada por palavras e por categorias.

A escolha da palavra “ERP” se deu pela abrangência do termo em relação aos aplicativos que se desejava obter e pela existência da categoria “ERP” no *site SourceForge.net*, objetivando assim manter um padrão quanto ao critério de busca, variando apenas as opções disponíveis no mecanismo de busca.

A Figura 16 demonstra como os resultados são apresentados pelo mecanismo de busca com o destaque para a existência da palavra “ERP” na descrição.

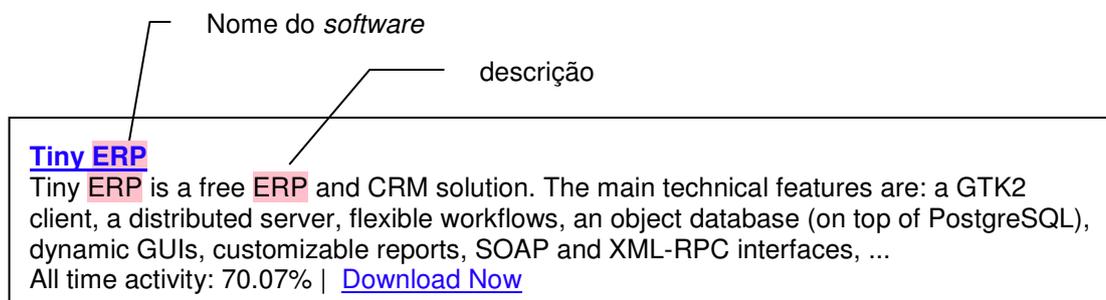


Figura 16 - Resultados do mecanismo de busca
Fonte: Adaptado de SourceForge.net (2005)

Foi essa experiência com o mecanismo de busca que tornou possível definir que os resultados obtidos poderiam ser conseguidos, por meio da combinação de buscas orientadas por “categoria” e por “palavra”. Conforme escrevem Blattmann e Tristão (1999): o pesquisador pode combinar diferentes formas, conforme a expectativa da informação que pretende obter.

Dessa forma, baseando-se na técnica de seleção de fases múltiplas e nas possibilidades encontradas no mecanismo de busca, as atividades para seleção das amostras foram divididas em cinco etapas:

Atividade 1: utilizar o mecanismo de busca do *SourceForge.net* com a opção de busca por “categorias”. Buscar pela categoria “ERP”.

Atividade 2: Utilizado o mecanismo de busca “por palavra” na descrição dos projetos. Buscar pela palavra “ERP”.

Atividade 3: Análisar os nomes dos projetos obtidos nos resultados dos dois tipos de pesquisa para eliminar possíveis duplicidades.

Atividade 4: Descobrir por meio de consulta ao cadastro do *software* se todos os projetos são realmente Sistemas Integrados de Gestão ou utilizam a palavra ERP por serem ferramentas ou aplicativos que tinham algum relacionamento com esta categoria de sistemas.

A consulta foi feita com base nas seguintes características:

- **Estágio do desenvolvimento:** esta variável é importante para determinar em que estágio de maturidade o projeto se encontra. O

próprio *SourceForge* possui parâmetros para isso que variam de 0 (não categorizado) a 5 (produção estável).

- **Área de aplicação:** caso o *software* fosse um sistema integrado de gestão, para atender mais especificamente a que área (indústria, comércio, serviço) ele foi concebido ?
- **Linguagem de programação:** qual é a linguagem de programação utilizada no código do programa?
- **Sistema operacional:** em quais tipos de sistema operacional é possível utilizar o programa?
- **Banco de dados:** com quais sistemas gerenciadores de bancos de dados o programa interage?
- **Hardware:** qual o tipo de computador necessário para o funcionamento do programa?
- **Idioma:** para quais idiomas o programa está disponível?

Atividade 5: utilizar a característica “**estágio de desenvolvimento do projeto**” para selecionar os aplicativos que foram instalados para a coleta de dados (preenchimento do questionário). Esta escolha foi feita porque esta variável indica quais projetos já estavam em estágio maduros o suficiente para utilização. Foram escolhidos os projetos de *software* que estavam no estágio de maior maturidade (5 – Produção estável), uma vez que já haviam superado inclusive a fase de testes que se dá no estágio anterior ao escolhido (4 – Beta).

É importante lembrar que a definição do estágio do projeto é feita de forma subjetiva e não tem relação com o modelo de maturidade da CMM (Capability Maturity Model) (REIS, 2003, p. 75).

Inicialmente havia se pensado em utilizar outro índice fornecido pelo repositório, o índice de vitalidade que indica a Taxa de atividade de Arquivos (*Files Activity %*). A taxa de atividade de arquivos representa a frequência com que os arquivos do projeto são visualizados. Reis (2003, p.75) chama-o de “índice de popularidade” e, segundo o autor, indica quantas pessoas se interessam pelo *software*. A característica “estágio de desenvolvimento do projeto” se mostrou mais adequada ao objetivo da pesquisa.

A Figura 17 representa a seqüência em que as atividades foram realizadas durante a seleção de amostras.

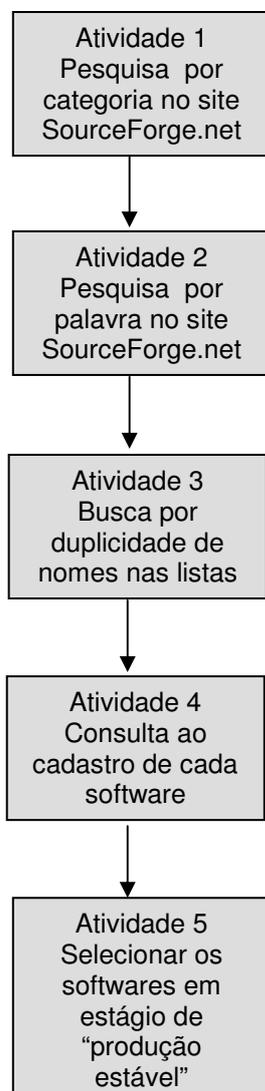


Figura 17 - Atividades da seleção de amostras

5.5 Instalação de aplicativos e coleta de dados

A instalação dos aplicativos foi feita no computador do Laboratório de Jogos de Empresas do Departamento de Engenharia de Produção da Escola de Engenharia de São Carlos. O equipamento é um micro computador com processador do tipo AMD - Duron 850 MHZ, com 256 Mb de memória RAM e um disco rígido com capacidade para 40 *Gigabites* e representa um tipo de *hardware* que é encontrado no ambiente das pequenas empresas.

Os aplicativos foram obtidos por meio das indicações do *SourceForge*. O repositório mantém uma área destinada a cada projeto para disponibilizar os aplicativos para *download*. As características de cada *software* instalado são descritas no capítulo 6.

A coleta de dados foi feita após a instalação de cada aplicativo.

5.6 Relacionamento com dados das pequenas empresas

O capítulo quatro deste trabalho apresentou dados e informações sobre as pequenas empresas coletados em instituições ligadas à esta categoria de empresas. O capítulo quatro foi construído baseando-se em equipamentos, pessoas e aplicativos utilizados no ambiente da pequena empresa.

No questionário formulado, existem questões referentes ao tipo de *hardware* requerido pelo aplicativo, dificuldades de instalação e características dos aplicativos. Com estas questões e com as informações obtidas no capítulo quatro foi realizado um comparação para evidenciar ou não a possibilidade de instalação dos aplicativos de *software* livre no ambiente da pequena empresa (apresentada na seção 6.5). Vale lembrar mais uma vez que este trabalho não trata do aspecto de implantação de sistemas, assunto que abrange muito mais do que a simples possibilidade de instalação.

5.7 Limitações

Algumas limitações do presente trabalho são inerentes ao próprio objetivo metodológico adotado. Quando Dane (1990) informa que os dados da pesquisa descritiva podem mudar com o decorrer do tempo, é justamente o que acontece nessa situação: o número de projetos de *software* livre está em constante crescimento, existem *softwares* em processo de desenvolvimento e a tecnologia está sempre se renovando. Não se pode garantir que as informações coletadas nesse estudo servirão como referência para um longo período de tempo, mas mesmo assim ainda podem servir como referência histórica para trabalhos futuros.

Outra limitação é imposta pelo próprio método de análise para a coleta de dados da terceira parte do questionário (usabilidade). Nielsen (1993,p.155)

comenta que a avaliação feita por um único avaliador pode deixar de verificar problemas na interface e informa que os problemas de usabilidade detectados por este método tendem a aumentar e as informações tendem a ser mais consistentes à medida que se aumenta o número de avaliadores. Em função do tempo para realização deste trabalho, bem como da complexidade de outros métodos que exigiram um trabalho exclusivo com foco apenas na usabilidade, o método Heurístico foi o escolhido.

No método descrito por Nielsen (1993), está intrínseca a observação do usuário e provavelmente diferentes tipos de usuários com conhecimentos e perfis diferentes podem interpretar as mensagens e diálogos apresentados pelos aplicativos de forma diferente. Para este trabalho foi adotada a visão de um tecnólogo em relação à mensagens, interface, documentos, enfim aos aspectos que constam do questionário.

Este capítulo apresentou a metodologia utilizada para a pesquisa referente ao seu referencial teórico, as variáveis de pesquisa, a construção do questionário utilizado, a maneira como foram selecionadas as amostras e as limitações do estudo. No capítulo seguinte, serão apresentados os resultados do estudo.

Capítulo 6 – Resultados

Este capítulo apresenta os resultados a partir da coleta e análise de dados. Os resultados estão divididos em duas fases: resultados gerais e *software* selecionados.

A primeira fase tornou possível a construção de um panorama geral dos sistemas integrados de informação gerencial desenvolvidos sob o modelo de código aberto. Na segunda fase, são descritos os *softwares* que passaram pela seleção imposta aos que foram coletados na primeira fase.

6.1 Fases da seleção de amostras

Uma vez definida a fonte dos dados (repositório de projetos *SourceForge.net*), utilizou-se o mecanismo de busca oferecido pelo próprio *SourceForge.net*. As atividades foram realizadas segundo a descrição apresentada na metodologia:

Atividade 1: busca utilizando a opção “por categorias”. Foi utilizada a categoria “ERP” e foi obtido um total de oitenta e três projetos.

Atividade 2: Foi utilizado o mecanismo de busca “por palavra” na descrição dos projetos. A palavra utilizada foi “ERP” e o retorno foi de duzentos e nove projetos.

Atividade 3: O próximo passo foi uma análise dos nomes dos projetos obtidos nos resultados dos dois tipos de pesquisa para eliminar possíveis duplicidades. Constatou-se que os oitenta e três projetos do resultado da busca por categorias estavam presentes na lista da busca por palavras, o que possibilitou a eliminação da primeira lista.

Atividade 4: Em seguida foi feita a consulta ao cadastro de cada aplicativo no repositório *SourceForge.net* e quando necessário e disponível consultou-se a página *web* do próprio projeto para a constatação de que o

projeto se tratava de um Sistema Integrado de Gestão ou havia utilizado a palavra ERP por ser uma ferramenta ou aplicativo que tinha algum relacionamento com esta categoria de sistemas. Esta consulta se baseou nas características estabelecidas na seção 2.3: estágio do desenvolvimento, área de aplicação, linguagem de programação, sistema operacional, banco de dados, *hardware* e idioma.

Observando os dados resultantes desta consulta, constatou-se que quarenta e sete aplicativos não pertenciam à categoria de sistemas integrados de gestão, sendo pertencentes aos outros tipos de sistemas relacionados a seguir:

- *Groupware*;
- Ambientes de desenvolvimento / *Framework*;
- Pacote para escritório;
- Módulos de ERP;
- Bibliotecas e componentes;
- CMS (*Content Management System*) – Gerenciador de conteúdo para páginas *web*;
- CRM (*Customer Relationship Management*).

Os outros cento e sessenta e dois projetos eram de *software* integrado de gestão empresarial. Uma lista detalhada é apresentada no Apêndice B.

Atividade 5: Utilizando a característica “estágio de desenvolvimento”, foram identificados quatorze sistemas em estágio de “produção estável” entre a lista dos cento e sessenta e dois. Estes quatorze *softwares* passaram pela fase de instalação para a obtenção dos dados do questionário. As características destes aplicativos são apresentados na seção 6.3. A Figura 18 representa a realização das tarefas durante a seleção de amostras, e o resultado obtido em cada uma das tarefas.

Quanto a estabelecer a significância desta amostra, é uma tarefa bastante difícil face à dificuldade de se obter o número exato de projetos de *software* livre existentes, uma vez que o registro não é obrigatório em qualquer repositório, embora seja bastante comum os interessados registrarem o projeto para que mais colaboradores possam participar. Também é possível encontrar o mesmo projeto cadastrado em mais de um repositório. Nota-se assim o

motivo da dificuldade em determinar a representatividade da amostra dentro do universo de milhares de projetos de *software* livre, porém ao considerar os *softwares* inscritos em *SourceForge* destinados a aplicativos de gestão integrada cujo estágio de desenvolvimento é considerado de “Produção estável”, esta amostra representa 100%.

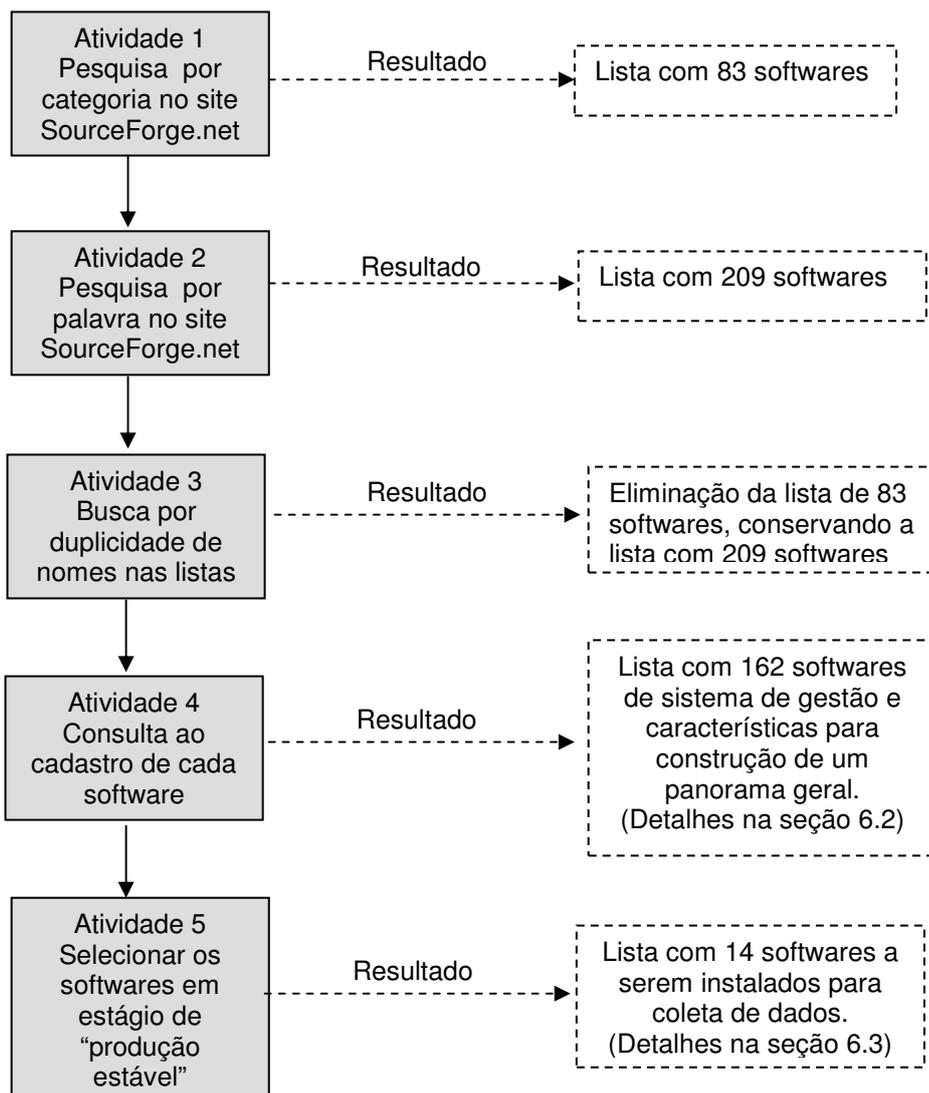


Figura 18 – Resultados das atividades da seleção de amostras

A próxima seção apresenta uma análise gráfica das características dos *softwares* selecionados até a atividade 4.

6.2 Panorama geral das características dos aplicativos

A obtenção dos dados dos aplicativos a partir do *site SourceForge.net* propiciou a possibilidade de construção de um panorama geral de suas características. A seguir são apresentados os detalhes dos resultados obtidos na fase de seleção de amostras e uma análise por meio da construção de gráficos com os dados obtidos obedecendo os critérios das características dos *softwares*.

6.2.1 Estágio de desenvolvimento dos projetos

O repositório *SourceForge.net* classifica os projetos em 6 diferentes estágios em relação à sua maturidade de desenvolvimento. Esta classificação obedece uma ordem crescente de 0 (zero) até 5 (cinco). Quanto maior o número, maior a maturidade do projeto. A Tabela 9 mostra as possibilidades de classificação do estágio do projeto. Para facilitar a compreensão dos dados durante a seleção (Apêndice B), foi criada uma sigla composta por duas letras.

Tabela 9 - Classificação de projetos segundo *SourceForge*

Sigla	Estágio do projeto segundo <i>SourceForge.net</i>
NC	Não Categorizado – 0
PN	Planejamento – 1
PA	Pré-Alpha – 2
AP	Alpha – 3
BT	Beta – 4
PS	Production/Stabile – 5

Apesar do número de projetos ser relativamente grande e o crescimento acontecer de forma rápida, quase um terço deles (30%) se encontra no estágio de “Não categorizado” o que significa, na prática, que o projeto é ainda uma idéia. Um terço deles (33%) ainda está em estágio de planejamento, 8% se encontram em fase de testes (Beta) quase prontos e aprovados para uso, e apenas 7% está maduro o suficiente para utilização em produção. O gráfico 7 ilustra esta situação.

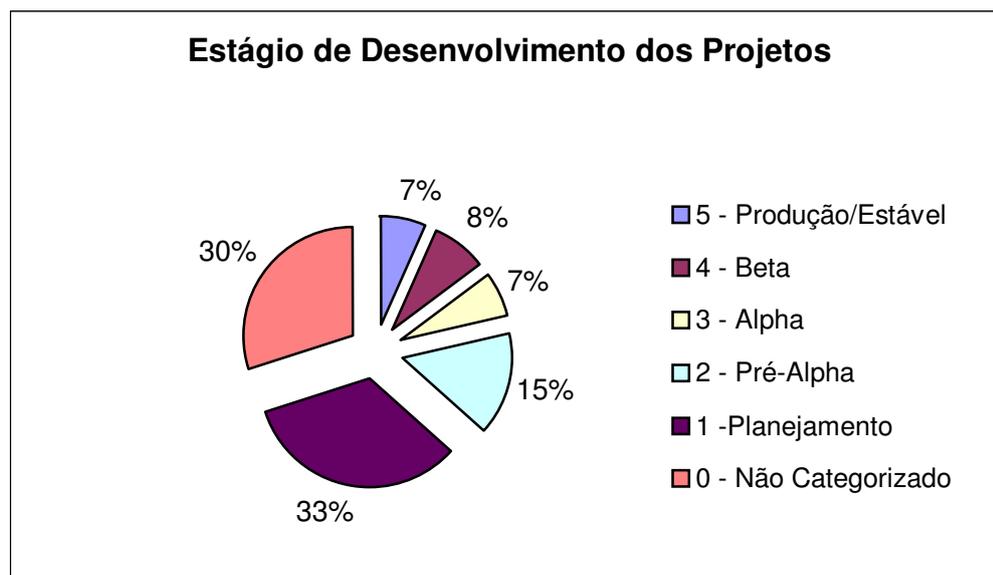


Gráfico 7 - Estágio de Desenvolvimento dos Projeto

6.2.2 Setor de aplicação

Foram identificados quatro diferentes setores: indústria, comércio, serviços e educação. Além destes foram também encontrados aplicativos que não declararam de forma explícita seu principal foco de atuação, sendo considerados “Não categorizados”. Foi criada uma sigla para identificar o setor a que o aplicativo pertencia. A Tabela 10 mostra esta classificação:

Tabela 10 - Classificação das áreas de aplicação do *software*

Sigla	Área de aplicação
IN	Indústria
CO	Comércio
SE	Serviços
ED	Educação
NC	Não Categorizado

Os aplicativos estão concentrados na aplicação à indústria apesar de não abranger totalmente todas as áreas que a atividade demanda, como por exemplo o setor de planejamento e controle da produção. Este fato foi evidenciado durante o estudo mais detalhado dos *softwares* escolhidos para o teste de instalação. No gráfico 8, é possível verificar a distribuição percentual dos setores em que os *softwares* podem ser utilizados.

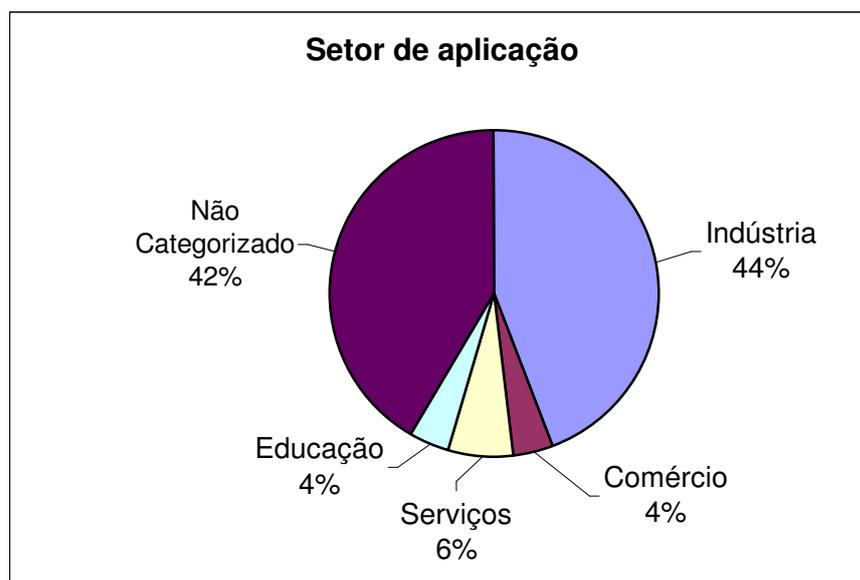


Gráfico 8 - Setores de aplicação

Pode-se notar que uma porção considerável (44%) se preocupa em direcionar o *software* para o setor industrial, mas uma quantidade praticamente igual (42%) não categorizaram o *software* como destinado à um setor específico, o que talvez aponte para uma possibilidade de utilização do aplicativo em diversos setores ou evidencia a possibilidade de adaptação do *software* livre às necessidades do usuário. Parcelas bem menores (entre 4% e 6%) dos aplicativos são destinados às áreas de educação, comércio e serviços.

6.2.3 Linguagens de programação

Com relação às linguagens de programação utilizadas para a construção dos códigos dos programas, percebe-se que elas não estão limitadas à uma linguagem especificamente, mas pode-se notar a preocupação em utilizar linguagens que também são desenvolvidas sob os moldes do *software* livre e que permitam a portabilidade, com destaque para Java (25,12%) e PHP (22,33%) que, juntas, são responsáveis pelo desenvolvimento de quase metade dos aplicativos. Vale lembrar que para as linguagem de programação é possível a existência de diferentes linguagens utilizadas para um único aplicativo.

A Tabela 11 apresenta as linguagens de programação utilizadas no desenvolvimento dos aplicativos com seus respectivos percentuais.

Tabela 11 – Linguagens de programação utilizadas

Linguagem Programação	%
Assembly	0,47
ASP	1,86
C	1,86
C#	2,79
C++	6,05
Delphi	2,79
Java	25,12
Java Script	5,58
JSP	0,47
Kylix	2,33
Object Pascal	0,47
Perl	3,26
Phyton	3,72
PHP	22,33
PL/SQL	1,86
Ruby	0,47
TCL	0,47
Unix Shell	1,40
Visual Basic	1,86
XSL	0,47
Não Categorizado	14,42

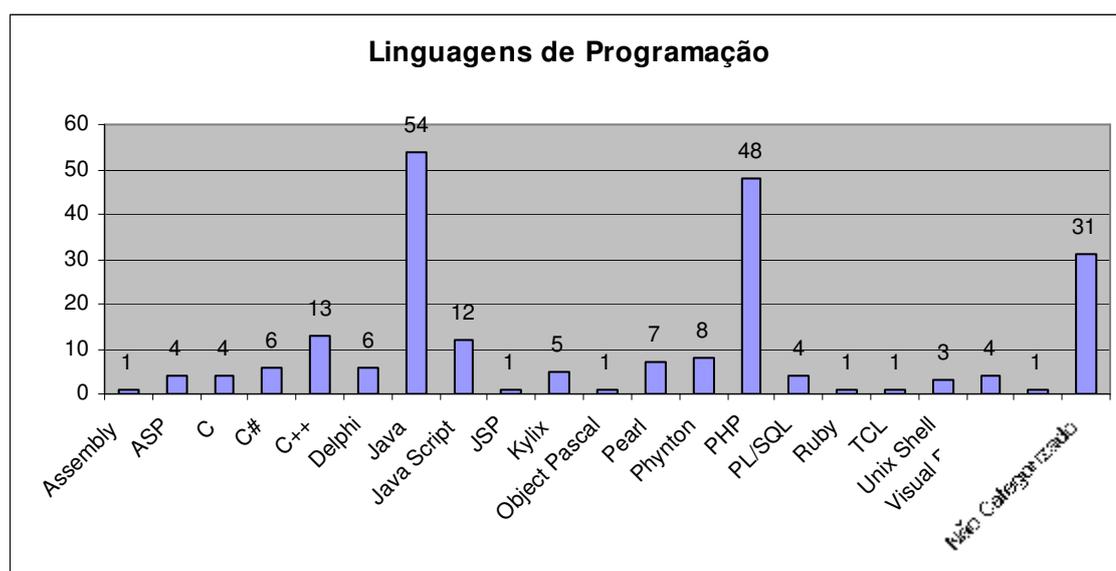


Gráfico 9 - Linguagens de programação utilizadas

No gráfico 9, é possível visualizar, em números de ocorrências, quanto cada linguagem de programação ocupa na distribuição. As ocorrências como

“não categorizado” aparecem em função daqueles projetos que ainda não definiram qual linguagem utilizarão.

6.2.4 Sistema operacional

Como reflexo da preferência da escolha por linguagens de programação que oferecem maior portabilidade os aplicativos têm buscado funcionar em diferentes sistemas operacionais, ou mesmo atender à todos os diferentes sistemas ao mesmo tempo, fator conseguido principalmente pela utilização do Java e PHP. Foi bastante comum encontrar cadastros de *software* onde o responsável pelo projeto declarava que o aplicativo estava apto a funcionar em Windows e POSIX (*Portable Operating System*), o que inclui sistemas como Linux e Unix. Mas a maioria deles (85 ocorrências) registrou que era possível utilizar o aplicativo em todos os sistemas operacionais. No gráfico 10, é possível visualizar as informações referentes ao sistema operacional de forma mais detalhada.

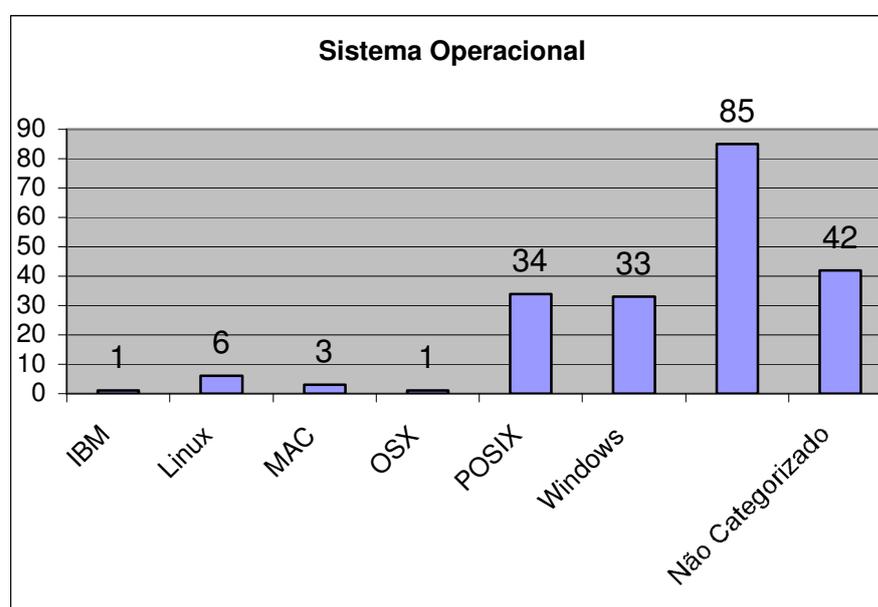


Gráfico 10 - Sistemas operacionais utilizados pelos aplicativos

Ainda com relação aos sistemas operacionais a que se destinam os programas, a Tabela 12 apresenta uma análise percentual da possibilidade de utilização dos sistemas integrados nos diversos sistemas operacionais.

Tabela 12 - Relação Percentual de utilização de Sistema Operacional

Sistema Operacional	Ocorrências	%
IBM	1	0,49
Linux	6	2,93
MAC	3	1,46
OSX	1	0,49
POSIX	34	16,59
Windows	33	16,10
Todos	85	41,46
Não Categorizado	42	20,49

6.2.5 Banco de dados

As informações referentes aos sistemas gerenciadores de banco de dados (SGBD) utilizados também segue o padrão de preferência por gerenciadores Open *Source* com destaque para o MySQL. A exceção à regra é o Oracle, que é um *software* proprietário, mas que por ocupar lugar de destaque no mercado desta categoria de sistemas acaba por criar uma necessidade de que sistemas integrados de gestão sejam capazes de se relacionar com este produto.

O Gráfico 11 apresenta a utilização dos diferentes gerenciadores de banco de dados.

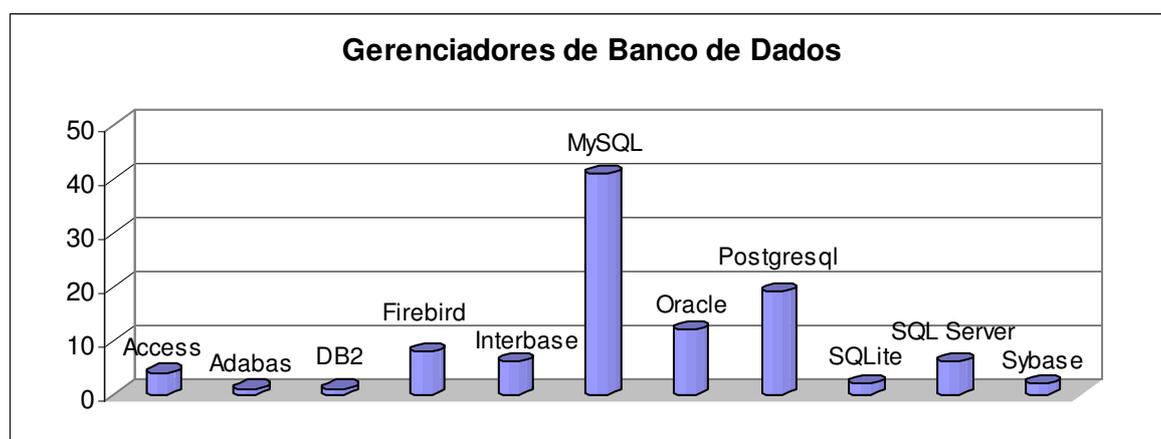


Gráfico 11 - Gerenciadores de Banco de Dados

Em alguns projetos, a informação encontrada no quesito banco de dados não foi propriamente o banco com que o aplicativo era capaz de interagir, mas

a forma utilizada para que isso acontecesse. Foram feitas referências à ferramentas que permitem que o aplicativo se conecte ao banco de dados, normalmente utilizando-se destas ferramentas para conseguir interagir com diferentes bancos, de acordo com a necessidade de quem utiliza o aplicativo. A Tabela 13 relaciona estas ferramentas e o número de vezes que foram citadas.

Tabela 13 - Ferramentas de acesso ao banco de dados

Ferramentas Acesso	Ocorrências
Ado DB	2
Database API*	1
Jboss	1
JDBC	11
MyODBC	1
ODBC	1
Pearl:DB	1
SQL Based	1
Web Logic	1
Todos os Bancos	1

6.2.6 Idiomas

Foram encontrados aplicativos desenvolvidos para atender a diversos idiomas. O fato do modelo de desenvolvimento utilizar pessoas de diferentes países, bem como a possibilidade de colaboração de diferentes pessoas nos projetos, mesmo como usuários, faz com que seja comum a possibilidade do aplicativo estar disponível em mais de um idioma. Na prática, esta característica nem sempre funcionou. Aplicativos como Compiere (versão da empresa Jotel) e Tiny ERP tinham a opção para utilização em português do Brasil, mas a escolha não funcionou, ou funcionou parcialmente com algumas partes da interface em português e outras em inglês.

Mesmo assim, é possível identificar que existe preocupação em produzir *software* para os usuários que falam português do Brasil quase na mesma proporção do italiano, alemão e francês. A preocupação com o português do

Brasil é maior do que com a de Portugal. A soma das ocorrências de português de Portugal e do Brasil ultrapassam o italiano, alemão e francês. O Gráfico 12 demonstra a diversidade de idiomas.

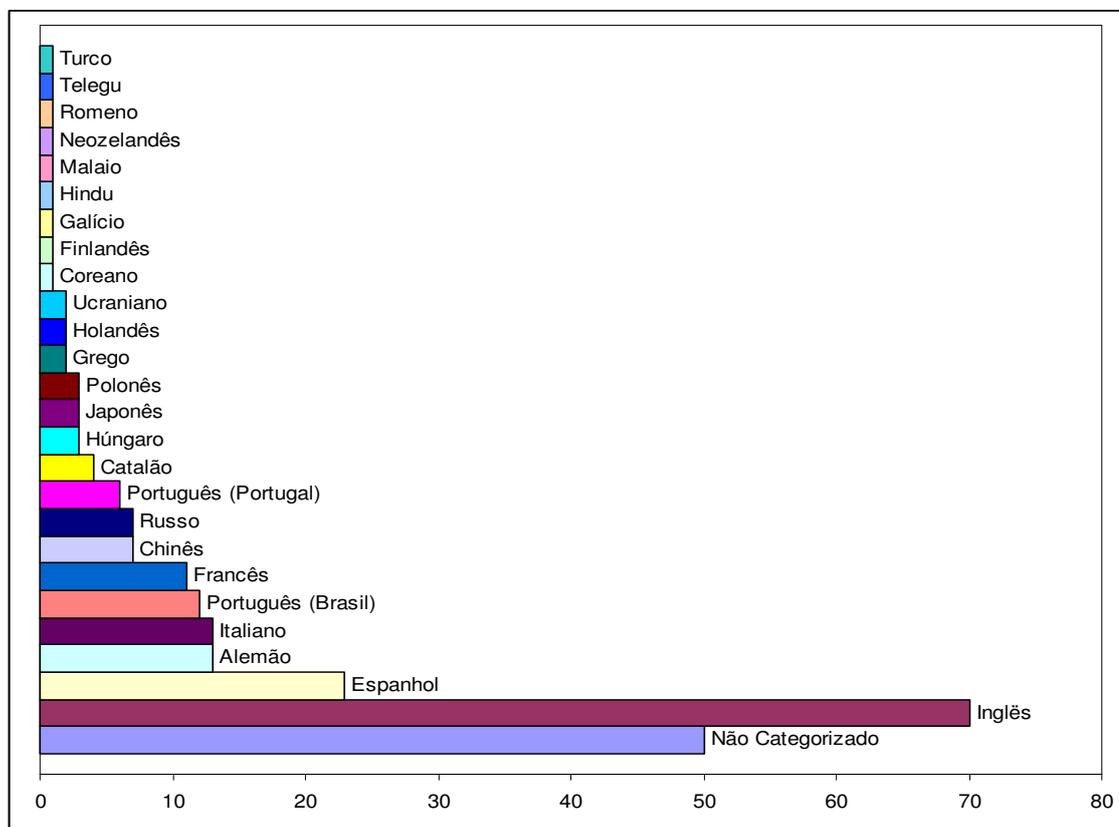


Gráfico 12 - Idiomas dos aplicativos

É importante notar que um aplicativo pode estar disponível em mais de um idioma. Os desenvolvedores trabalham com os arquivos de identificação e mensagens em separado do código de desenvolvimento escrito em linguagem de programação, de forma que é possível mesmo àqueles que não sabem programar poderem auxiliar na tradução deste arquivo para outras línguas. A Tabela 14 apresenta os idiomas e os números de ocorrências de cada um deles.

Tabela 14 - Número de ocorrências de idiomas

Idioma	Ocorrências
Não Categorizado	50
Inglês	70
Espanhol	23
Alemão	13
Italiano	13
Português (Brasil)	12
Francês	11
Chinês	7
Russo	7
Português (Portugal)	6
Catalão	4
Húngaro	3
Japonês	3
Polonês	3
Grego	2
Holandês	2
Ucraniano	2
Coreano	1
Finlandês	1
Galício	1
Hindu	1
Malaio	1
Neozelandês	1
Romeno	1
Telegu	1
Turco	1

Assim, foram apresentados os resultados das características dos *software* de forma geral. A próxima seção vai descrever as características dos *softwares* selecionados para instalação.

6.3 Softwares selecionados

Conforme já foi descrito, a característica “estágio de desenvolvimento do projeto” foi determinante na escolha dos aplicativos que foram instalados para a coleta de dados. Foram encontrados quatorze *softwares* em estágio de “produção estável”. O quadro 10 apresenta estes aplicativos.

	Nome Sistema	Área aplicação	Estágio do projeto	Linguagem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
1	Compiere	IN	PS	Java	Linux Windows MAC	Oracle, Postgresql, Firebird	PC / MAC	Inglês, Catalão, Alemão, Espanhol, Francês, Finlandês, Italiano, Japonês, Neozelandês, Polonês, Português (Brasil), Russo, Chinês, entre outros
2	Evaristo	IN	PS	Java	Todos	JDBC*	PC	Português (Portugal)
3	EzyBiz	IN	PS	PHP	Todos	MySQL	PC	Inglês
4	Factura LUX	IN	PS	C++	Linux, Windows,MAC		PC	Espanhol Português
5	Freedom ERP	IN	PS	Java	Todos	Firebird, Interbase, JDBC	PC	Português (Brasil)
6	Linux-Kontor	IN	PS	Java	Linux, Windows, MAC	Adabas, Postgresql	PC	Alemão, Inglês
7	Microerp ERP System	IN	PS	Java	Todos	JDBC*	PC	NC
8	openXpertya	IN	PS	Java, JavaScript	Todos	Oracle	PC	Português (Brasil), Catalão, Galicia, Português, Espanhol
9	Oratio	NC	PS	Perl, PL/SQL	Window, POSIX	Postgresql, Oracle,	NC	□íngua, □íngua□, Grego, Italiano, Espanhol
10	Sequóia ERP	NC	PS	Java	Windows, POSIX	DB2, SQL Server, MySQL, Oracle, Postgresql	PC	Chines (Simplificado), Inglês, Frances , Alemão , Italiano, Espanhol
11	Tiny ERP	NC	PS	Python XSL	Todos	SQL-based	PC	Alemão, Inglês, Francês, Espanhol
12	WebERP	IN	PS	PHP	Todos	MySQL, Postgresql	PC	Espanhol, Inglês, Polonês, Português (Brasil), Russo
13	University ERP System	ED	PS	Java	Todos	MySQL, Oracle	PC	Inglês, Malaio
14	Value	IN	PS	Delphi/ Kylix, Java	Windows, POSIX	MySQL, Oracle, Postgresql	PC	Inglês

Quadro 10 – Aplicativos selecionados

Estes aplicativos podem ser divididos em grupos, obedecendo o aspecto das linguagens de programação adotadas para sua codificação. O quadro 11 apresenta esta divisão.

Java	PHP	C++	Python	Perl
Compiere Evaristo Freedom ERP Linux Kontor MicroERP openXpertya Sequóia ERP University	Ezybiz WebERP Value	FacturaLux Lite	TinyErp	Oratio

Quadro 11 - Aplicativos selecionados segundo a linguagem de programação

6.3.1 Pré-requisitos para instalação

Os aplicativos que foram escritos na linguagem PHP necessitam que seja instalado um servidor de páginas *web* para que possa ser visualizada a interface do sistema. O PHP é uma linguagem interpretada que permite a realização de cálculos, utilização de variáveis e interação com banco de dados mas que não possui recursos para a construção de interfaces com o usuário. Não se criam objetos como espaços para se digitar texto ou janelas de diálogo com os recursos da linguagem PHP. O uso mais comum do PHP é feito em conjunto com a linguagem HTML que permite a criação dos componentes da estrutura para interação do usuário com o sistema.

Um dos servidores de páginas *web* mais utilizado é o aplicativo chamado Apache que já foi citado como um exemplo de projeto de *software* livre bem sucedido, juntamente como o sistema operacional Linux.

Também é bastante comum a utilização de outro *software* livre, o gerenciador de banco de dados MySQL em conjunto com Linux, Apache e PHP. Tão comum que a integração destes quatro programas acabou sendo designada pela sigla: LAMP (**L**inux – **A**pache – **M**ysql – **P**HP, **P**erl ou **P**ython). A partir da definição da sigla, assume-se que o ambiente é necessário também às linguagens Perl e Python.

Segundo Wikipédia (2005) LAMP ou L.A.M.P é um acrônimo usado para referenciar o conjunto de programas *open source* que interagem para propiciar um ambiente *web* que possibilite a utilização de páginas dinâmicas.

Quando o sistema operacional Linux é substituído pelo Windows a letra L do acrônimo é substituída pela letra W, gerando assim o acrônimo WAMP (**W**indows – **A**pache – **M**ysql – **P**HP, **P**erl ou **P**ython). Da mesma forma o acrônimo MAMP é válido para **M**achintosh – **A**pache – **M**ysql – **P**HP, **P**erl ou **P**ython.

Por isso quando algum projeto mencionou a necessidade da plataforma LAMP / WAMP / MAMP significava que a estrutura necessária para seu funcionamento dependia da existência destes outros aplicativos.

Para este trabalho o termo utilizado na descrição dos aplicativos para se referir à esta plataforma será AMP, uma vez que tanto Apache como MySQL ou PHP funcionam em diferentes sistemas operacionais.

Os aplicativos que foram instalados e que se utilizam destes recursos são os seguintes: webERP, Ezybiz e Oratio.

Como facilitador da instalação dos três aplicativos (Apache, MySQL e PHP) é possível encontrar disponível na internet programas que se encarregam da instalação e configuração destes outros programas, deixando o ambiente pronto para a instalação do aplicativo integrado de gestão ou qualquer outro aplicativo que se utilize da plataforma AMP. Um exemplo destes instaladores é o XAMP disponível no próprio *site* do servidor Apache e que foi utilizado para os testes de instalação.

Os aplicativos desenvolvidos utilizando a linguagem Java também precisam que seja instalado seu interpretador. Este aplicativo tem diferentes versões cuja necessidade precisa ser identificada. Este interpretador, assim como a própria linguagem Java, foram desenvolvidos pela empresa *Sun Microsystem* e se encontram disponíveis no *site* do próprio fabricante.

A instalação da “máquina virtual”, conforme é tecnicamente chamado o interpretador da linguagem Java, ocorre de forma simples uma vez que o usuário da instalação se encarrega desta tarefa, bastando que o usuário configure o local de instalação.

Os aplicativos que utilizam as linguagens C++ e Delphi para desenvolvimento do código dispensam a instalação de interpretadores pois já

foram transformados em um programa que pode ser diretamente executado pelo sistema operacional. São tecnicamente conhecidos como programas executáveis.

Com respeito ao gerenciador de banco de dados, os aplicativos instalados faziam uso do Postgresql ou MySQL. O Postgresql possui um instalador para o sistema operacional Windows, mas só pode ser utilizado nas versões XP, 2000 ou NT deste sistema operacional. As versões anteriores não suportam o Postgresql por uma questão técnica quanto à forma de gerenciamento do espaço de memória RAM e gerenciamento do espaço em disco rígido. Para uso no sistema operacional Linux também existe um aplicativo para auxiliar na instalação ou então a instalação pode ser feita de forma manual sem restrições quanto à versão específica. Por uma questão de restrição de tempo, foram mantidas a forma de acesso ao banco de dados que estavam estabelecidas por padrão quando foi feito o *download* do aplicativo, ou seja, mesmo que um aplicativo pudesse fazer uso de mais de um tipo de gerenciador de banco de dados (possibilidade bastante comum) ele foi instalado apenas com um deles.

Os gerenciadores de banco de dados utilizam uma linguagem específica para manipulação dos dados chamada de SQL (Structured Query Language) permitindo a inserção de dados ou criação de tabela a partir de um arquivo texto escrito neste padrão. Este trabalho trata este tipo de arquivo com o nome de *.sql.

As seções a seguir descrevem individualmente cada um dos sistemas instalados.

6.3.2 Compiere

Conforme Saleh (2003), o Compiere é um aplicativo de gestão empresarial desenvolvido originalmente sob encomenda da Goodyear da Alemanha. A primeira versão data de 2000. O objetivo do aplicativo é atender as necessidades de pequenas e médias empresas (COMPIERE, 2003).

O Compiere apresentou-se como o mais completo e maduro dos sistemas testados atendendo às diferentes áreas da empresa de forma completa. Embora não tenha todos os módulos projetados totalmente

desenvolvidos, pretende atender recursos de cadeia de suprimentos (supply chain) e relacionamento com clientes (CRM) .

A indicação de que este aplicativo tem credibilidade é a existência de diversas empresas que trabalham como parceiras oferecendo serviços de consultoria e adequação do software à necessidade de seus usuários. No Brasil, duas empresas oferecem estes serviços: a Visuelles Informática e a Conceptia *Consulting*.

Estas empresas normalmente promovem adequações de idioma e regras de leis fiscais e contábeis para que o sistema esteja apto à utilização em determinado país ou região.

A versão original do Compiere apresentou um empecilho quanto ao propósito deste trabalho em relação à sua instalação: o gerenciador de banco de dados utilizado para esta versão é o Oracle que se trata de um software proprietário. Embora exista uma versão do Oracle disponível para estudos, possibilitando testes, ela não poderia ser utilizada por uma empresa sem a aquisição de licenças.

Existe, em desenvolvimento pela equipe do Compiere, uma atualização para que seja possível utilizá-lo com outros bancos de dados, mas o trabalho ainda não está terminado. Foi necessário buscar alternativas e foram encontradas outras versões desenvolvidas que podem funcionar com um banco de dados que esteja na categoria de *Open Source*. Algumas versões foram testadas:

- Versão distribuição (*Compiere Libero*) da empresa mexicana *Eevolution*, que tem características para utilizar o Servidor de Banco de Dados Postgresql. Foram encontradas divergências entre as versões do interpretador Java para esta versão e da versão original. Foi necessária a desinstalação da versão 1.4.5 do Java e instalação da versão 1.4.2. Depois de instalado foram encontradas algumas dificuldades com a conexão com o banco de dados.
- Versão distribuída pela empresa Jotel, uma empresa da Polônia e utiliza como banco de dados o Fyracle. Este banco de dados é uma versão do banco Firebird com a possibilidade do uso dos recursos das funções do Oracle. Esta adaptação no banco de

dados foi realizada pela empresa holandesa chamada Janus Software. Para esta versão realizou-se a instalação completa, embora ao iniciar a utilização fosse emitida uma mensagem de que seria necessária uma atualização no banco de dados.

- Versão distribuída pela empresa brasileira Conceptia. Esta empresa recebeu recursos de um projeto da Finep (Financiadora de estudos e projetos) para tornar possível a utilização do Compiere com o banco de dados Postgresql e a tradução para o Português. Foi sobre as características desta versão que os dados foram coletados.

O Compiere apresenta como características o fato de poder gerenciar mais de uma empresa, permite controle em diferentes padrões de contabilidade e permite diferentes métodos de custos de estoque (média, preço padrão, PEPS).

O sistema também foi projetado para atender a diferentes idiomas, embora ainda não o faça completamente. Nas versões instaladas, apesar da existência da opção idioma Português, somente a versão desenvolvida no Brasil é que trouxe efetivamente a implementação desta característica. Estão disponíveis os idiomas: inglês, chinês tradicional e simplificado, espanhol, suíço, francês, italiano, alemão, russo, dinamarquês, indonésio e malaio.

A instalação deste aplicativo demanda algum conhecimento em informática pois é necessário que sejam configuradas algumas variáveis de sistema, conhecidas como variáveis de ambiente, para que o sistema possa funcionar corretamente, além da configuração do próprio sistema que necessita de conhecimentos de rede de computadores. Também se faz necessária a instalação do Java juntamente com as características de desenvolvimento que esta linguagem oferece o JDK (*Java Development Kit*), bem como instalação do banco de dados em acordo com a versão escolhida do sistema. Tudo isto para a instalação do servidor de aplicação, pois caso seja necessário a instalação em estações de trabalho, outras configurações são requeridas. Estes detalhes fizeram com o nível de dificuldade de instalação fosse considerado alto.

O Compiere funciona a partir da execução de uma programa servidor que permite às estações de trabalho fazerem uso da aplicação tanto por meio da utilização de um navegador *web*, quanto por uma aplicação cliente instalada na estação de trabalho. Vale dizer que para o acesso por meio do navegador *web* não existe a disponibilidade do idioma em Português para nenhuma das versões instaladas.

As exigências de *hardware* deste aplicativo estão restritas aos equipamentos que suporte os sistemas operacionais nos quais ele pode ser instalado (Windows, Linux, BSD e outros Unix *like*) e funcionou de forma satisfatória no equipamento utilizado para testes (AMD Duron 850 Mhz com 256 Mb de memória RAM).

A licença do Compiere é baseada na MPL (Mozilla Public Licence) o que faz do Compiere um aplicativo do tipo Open Source.

Diversos relatórios são possíveis de serem extraídos e podem ser visualizados na tela do computador ou enviados à impressora. Existe também a possibilidade de exportar os relatórios em vários formatos de arquivos entre eles o PDF (*Portable Document Format*) e o *xml* (*eXtensible Markup Language*), além de recursos que permitem que sejam gerados gráficos com as informações disponíveis no sistema.

O acesso ao aplicativo é permitido por uma senha e o administrador do sistema pode criar diferentes usuários e diferentes regras de permissão de acesso e uso do sistema para cada usuário.

A interface do sistema possui linguagem clara e oferece ao usuário mensagens de erros de fácil interpretação. Existe documentação sobre instalação disponível e a documentação de utilização do sistema fica por conta de um mecanismo de ajuda que explica cada um dos campos a serem preenchidos na tela do sistema (na versão desenvolvida no Brasil, o texto de ajuda está em Português). Documentação mais detalhada pode ser conseguida por meio da compra de manuais. O próprio “desenvolvedor original” do sistema disponibiliza uma “loja virtual” onde pode ser adquirido material além dos manuais, vídeos e cursos. O desenvolvedor oferece também um programa de capacitação e certificação para profissionais e empresas parceiras. Toda a documentação e os cursos oferecidos são em inglês.

No Brasil, as empresas desenvolvedoras oferecem suporte. Também é possível conseguir informação nas listas de discussão.

O quadro 12 mostra informações sobre o Compiere.

Nome: Compiere	
Web site: http://www.compierebrasil.com.br (versão brasileira Conceptia) http://www.compiere.com.br (versão brasileira Vesuelles) http://www.compiere.org (versão desenvolvedor original)	
Sistema operacional: Windows, Linux, BSD Tipo de licença: MPL e Unix <i>like</i>	
Nº inscritos grupo discussão: possui 15 grupos divididos em diferentes assuntos	
Indicação hardware: equipamento que suporte Windows, Linux ou Unix	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Alta	
Software adicional: Java to Standard Edition com Java Development Kit (JDK) Environment e banco de dados de acordo com a versão (Oracle, Postgresql, Fyrcle)	
Banco de dados nativo: Oracle	
Idioma – disponível em português: Sim (para versão feita no Brasil)	
Gerencia mais de uma empresa: Sim	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Sim
Possibilita integração outro software: Não	Ferramentas comunicação: Sim
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim

Quadro 12 - Informações Compiere

6.3.3 Evaristo

Utilizando a licença do tipo GPL, o Evaristo é um software cujo desenvolvedor declara ter sido criado para atender principalmente as pequenas indústrias. Como requisitos de *hardware* são indicados computadores que suportem a instalação do sistema operacional NT, Linux ou Mac para o funcionamento do banco de dados. A instalação indica a utilização do banco de dados Postgresql, mas o sistema pode também fazer uso do Oracle, DB2, SQL Server, e outros, porque é utilizada uma ferramenta chamada JDBC (Java Database Connectivity) que permite a conexão com diferentes tipos de gerenciadores de banco de dados.

Este aplicativo não possui um instalador e a instalação deve ser feita por meio da descompactação dos arquivos, criação de um usuário específico para o banco de dados e criação da base de dados a partir de arquivo do tipo *.sql.

Estes passos de instalação fazem com que o nível de dificuldade de instalação seja considerado médio.

As alterações em tabelas do banco de dados e na interface do programa só podem ser feita mediante a alterações do código-fonte do sistema.

Os relatórios oferecidos atendem as áreas de compras, vendas e produtos. São oferecidos relatórios por fornecedor, por produto, por cliente, resumo de documentos, detalhe de documentos, vencimentos e avisos de crédito, além de movimento de produtos. Os relatórios podem ser visualizados na tela do computador ou enviados à impressora, mas não existe uma opção para transformá-los em arquivos.

Não existe, presente no Evaristo, a possibilidade de integração com outros softwares nem a presença de ferramentas de comunicação próprias para envio de mensagens a outros usuários.

O Evaristo não propicia uma estrutura que permita o uso de senhas para acesso ao sistema, nem o gerenciamento de diferentes tipos de usuários, ficando a segurança restrita ao gerenciador de banco de dados e ao sistema operacional.

Com relação ao *backup*, o Evaristo disponibiliza uma opção para a realização da cópia de segurança a partir do próprio *software*. O quadro 13 disponibiliza informações sobre o *software* Evaristo.

Nome: Evaristo	
Web site: http://www.m16e.com	
Sistema operacional: Todos	Tipo de licença: GPL
Nº inscritos grupo discussão: não possui	
Indicação hardware: equipamento que suporte Windows NT, Linux ou Mac para servidor de banco de dados e a partir do Windows 95 para estações	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Média	
Software adicional: Java to Standard Edition Runtime Environment e Postgresql	
Banco de dados nativo: Postgresql	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Não	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Não	Diferentes níveis de usuário: Não
Backup interno: Sim	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fácies e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Não
Documentação disponível : Sim	Suporte: Sim

Quadro 13 – Informações Evaristo

Durante a verificação da usabilidade, percebeu-se a preocupação com o uso de objetos, mensagens e botões de forma clara e simples, porém os módulos de produtos e vendas apresentaram problemas de instalação e não puderam ser testados.

6.3.4 EzyBiz

Com início de desenvolvimento em 2004, este software não possui grupo de discussão cadastrado em SourceForge.net, mas está em estágio de produção estável.

O EzyBiz foi projetado para funcionar em todos os sistemas operacionais. Sua licença é não-permissiva: GPL.

Este aplicativo necessita da instalação da plataforma AMP para seu perfeito funcionamento e indica também a necessidade de 64Mb de memória RAM e 160 Mb de espaço em disco rígido.

A instalação é feita mediante a descompactação dos arquivos na pasta do servidor *web*. É necessário ajustar a versão do PHP para 4 e criar as tabelas no banco de dados por meio da importação dos arquivos *.sql distribuídos juntamente com os arquivos de instalação. Também é necessário configurar as permissões de usuários em determinadas pastas. Estes detalhes fazem com que a instalação tenha sido considerada como tendo um nível médio de dificuldade.

O gerenciador de banco de dados com o qual o EzyBiz trabalha é o MySQL, não aceitando nenhum outro.

É possível importar dados para o sistema por meio de arquivos no formato *.sql ou *.csv (dados em forma de texto separados por vírgula).

O EzyBiz permite o gerenciamento de dados de mais de uma empresa no mesmo banco de dados, possibilitando integração, por exemplo, entre matriz e filial. As possíveis alterações necessárias na interface ou na base de dados precisam ser feitas diretamente nas tabelas e no código fonte do programa.

Os relatórios gerados pelo aplicativo podem ser visualizados na tela do computador e impressos por meio dos recursos de impressão do navegador *web*. Existe a disponibilidade de relatórios de pedidos, contas, estoque,

compras, folha de pagamento, livro caixa e atividade de projetos. Também é possível gerar gráficos para o livro caixa.

O quadro 14 apresenta informações sobre o Ezybiz.

Nome: Ezybiz	
Web site: http://ezybiz.sourceforge.net	
Sistema operacional: Todos	Tipo de licença: GPL
Nº inscritos grupo discussão: não possui	
Indicação hardware: 64 Mb RAM e 160 MB disco rígido	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Média	
Software adicional: plataforma AMP	
Banco de dados nativo: MySQL	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Sim	Possui relatórios: Sim
Exporta relatórios em arquivos: Não	Gerador de gráficos: Sim
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Não
Indicações de resultados e controles: Não	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Não	Solução dúvidas utilização: Não
Documentação disponível : Sim	Suporte: Não

Quadro 14 - Informações Ezybiz

A exemplo de outros aplicativos, o EzyBiz não possui recursos de integração com outros softwares nem ferramentas próprias de comunicação, como *e-mail* ou envio de mensagens à outros usuários.

É possível controlar diferentes níveis de acesso para os usuários do EzyBiz e determinar se o usuário pode apenas consultar ou alterar dados, mas não existe ferramenta de *backup* dos dados integrada ao software.

Quanto ao idioma, o EzyBiz está disponível somente em inglês e também não apresenta um mecanismo de solução de dúvidas de usuários.

Botões, objetos, opções e mensagens não são apresentadas de forma clara e de fácil compreensão, embora a linguagem utilizada em menus e diálogos utilize expressões comumente encontradas nos outros aplicativos deste tipo. Estas falhas aliadas à inexistência de um grupo de discussão e de outras formas de suporte à instalação (como empresas que oferecem suporte) constituem o ponto fraco do EzyBiz.

6.3.5 FacturaLux Lite

Este aplicativo teve seu início de desenvolvimento em 2001. A linguagem de programação utilizada é o C++, e o sistema se encontra disponível para os sistemas operacionais Linux, Windows e Mac.

A indicação do desenvolvedor do hardware necessário para funcionamento do sistema é bastante modesta: um microcomputador do tipo Pentium 133 Mhz, com 256 MB de memória RAM e 1 GB de espaço em disco.

O FacturaLux faz uso da licença GPL. O aplicativo possui um instalador que facilita sua instalação e caso seja necessário oferece também a opção de desinstalação do sistema.

Quanto ao gerenciador de banco de dados, o FacturaLux é configurado durante sua instalação para funcionar com o SQLite e não precisa da instalação prévia deste para funcionar. Tudo é feito durante o processo de instalação, fazendo com que a dificuldade de instalação seja considerada baixa.

Caso seja de preferência do usuário, é possível fazer com o FacturaLux utilize o Postgresql. Neste caso é necessário providenciar a instalação deste gerenciador, conforme descrito na seção 6.3.1

Não existe recursos para a importação da base de dados que existia na empresa antes da instalação do aplicativo. Isto só é possível mediante a transformação dos dados existentes no formato de arquivo *.sql.

Também não existem recursos internos para adequação de dados nas tabelas e ajustes na interface do software. Estes detalhes, quando necessários, devem ser alterados por meio da alterações do código fonte do sistema e da manipulação direta da base de dados.

O FacturaLux permite que sejam gerenciadas mais de uma empresa, mas não permite a integração entre os dados das diferentes empresas, pois o controle de mais de uma empresa se dá mediante a criação de um novo banco de dados.

Relatórios referentes à Livro caixa, balanços, faturas recebidas e emitidas, vendas, compras e inventário estão disponíveis no ambiente do FacturaLux e podem ser visualizados tanto na tela do computador, quanto

enviados à impressora. Não existe o recurso de transformação dos relatórios em arquivos nem ferramentas para a criação de novos relatórios. Também não existe recurso presente no software para gerar gráficos.

Recursos de integração com outros softwares e ferramentas próprias de comunicação não estão presentes no FacturaLux.

Uma deficiência do FacturaLux é não oferecer possibilidade de criação de senha ou diferentes níveis de acesso ao aplicativo, a não ser que seja adquirido um módulo à parte, que deve ser pago.

No quadro 15 estão sintetizadas informações sobre o FacturaLux.

Nome: FacturaLux Lite	
Web site: http://facturalux.org	
Sistema operacional: Linux, Windows, Mac	Tipo de licença: GPL
Nº inscritos grupo discussão: não possui	
Indicação hardware: Pentium 133 MHZ com 256 Mb memória RAM e 1Gb espaço em disco	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Sim	Opção de desinstalação: Sim
Dificuldade de instalação: Baixo	
Software adicional: Não requerido	
Banco de dados nativo: SQLite	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Não	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Não	Diferentes níveis de usuário: Sim (módulo pago)
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim (empresas prestadoras de serviço, mas nenhuma no Brasil)

Quadro 15 – Informações FacturaLux

Os requisitos de usabilidade fazem parte do aplicativo que apresenta mensagens claras e uma interface com botões, objetos e opções apresentadas de forma fácil e visível, além de um mecanismo de ajuda e solução de dúvidas quanto à utilização do *software*, bem como documentação detalhando a instalação e utilização.

Existe um problema com relação ao idioma. Embora exista a indicação da disponibilidade do aplicativo em espanhol, português e alemão, somente o espanhol pode efetivamente ser instalado.

Com relação ao suporte, existem empresas atuando na manutenção, instalação, migração de dados e cursos para o FaturaLux, porém nenhuma delas atua no Brasil.

6.3.6 Freedom

O Freedom é um sistemas desenvolvido por brasileiros. Utilizando-se da linguagem Java, o aplicativo foi projetado para atender à indústria e ao comércio.

O sistema faz uso do banco de dados Firebird e não exige recursos sofisticados de hardware para funcionar. O Freedom apresenta possibilidade de utilização dos módulos de forma separada, mas integrada à uma base de dados única. Por exemplo, o departamento financeiro pode executar apenas o módulo do sistema referente ao controle das finanças, mas existe também a opção de utilização de um ambiente com todos os módulos. Existem módulos para a área financeira, produção, ponto de venda, telemarketing e atendimento ao cliente.

O Freedom permite o gerenciamento dos dados de mais de uma empresa. Os relatórios podem ser visualizados na tela do computador ou enviados à impressora. É possível também exportar os relatórios em formato de arquivo PDF ou texto simples (TXT), mas o sistema não disponibiliza uma ferramenta para gerar gráficos com as informações.

Não existem ferramentas integradas de comunicação nem a opção de cópia de segurança integrada ao sistema.

É necessário uma senha para acesso ao sistema e existe a presença de um mecanismo que permite a criação de diferentes usuários com diferentes permissões de acesso ao sistema.

A interface oferecida pelo Freedom é intuitiva e amigável, utilizando-se de uma linguagem simples e dispondo objetos de forma visível e de fácil acesso. O sistema disponibiliza formas mais rápidas de acesso à opções do sistema (atalhos) para usuários mais experientes e as mensagens de erros são apresentadas de forma clara.

O aplicativo está disponível apenas no idioma Português (Brasil). No quadro 16 pode-se encontrar informações referentes ao aplicativo Freedom.

Nome: Freedom	
Web site: http://freedom.org.br	
Sistema operacional: Windows, BSD, Linux, Tipo de licença: GPL OSX	
Nº inscritos grupo discussão: 3	
Indicação hardware: equipamento que suporte sistema operacional escolhido	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Baixo	
Software adicional: Java to Standand com Runtime Environment e Firebird	
Banco de dados nativo: Firebird	
Idioma – disponível em português: Sim	
Gerencia mais de uma empresa: Sim	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fácies e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Não	Suporte: Sim (por <i>e-mail</i> e pela empresa desenvolvedora. Disponível no Brasil)

Quadro 16 - Informações Freedom

A dificuldade de utilização do Freedom está por conta da falta de documentação disponível sobre a instalação e documentação do sistema, que apesar de apresentar um baixo nível de dificuldade torna-se dificultada à um leigo justamente pela falta de documentação. A falta de informações prejudica também o primeiro acesso ao sistema. Para que fosse possível sua utilização foi necessário o envio de *e-mail* ao desenvolvedor para a obtenção da senha de acesso ao sistema. A referida senha era a mesma do usuário padrão do banco de dados Firebird.

O sistema Processa é outro *software* nacional de gestão de empresas que foi localizado durante a época da pesquisa, mas que não se encontrava cadastrado no *site* SourceForge e por isso não faz parte da amostra. Outro aplicativo é o ERP5 (um *framework*) que embora não tenha o núcleo de desenvolvimento no Brasil, tem a participação do CEFET Campos, localizado no Paraná, como parceiro em seu desenvolvimento.

6.3.7 Linux Kontor

Linux Kontor é um software que utiliza o Java como linguagem de desenvolvimento. O cadastro do aplicativo no *site* SourceForge aponta para um estágio de desenvolvimento estável e disponibilidade nas linguagens alemão e inglês.

Infelizmente o pacote disponível para *download* não continha informações sobre como proceder para que o mesmo fosse instalado.

Foram feitas tentativas de localização de maiores informações e documentação sobre o sistemas, mas a *home page "oficial"* indicada no cadastro do aplicativo também não estava funcionando.

Uma análise nos comentários do código fonte revelou que os mesmos foram escritos utilizando o alemão como idioma. Este conjunto de dificuldades acabou por impossibilitar a instalação deste aplicativo, uma vez que o autor deste trabalho não têm domínio do idioma alemão.

Por não ter sido instalado, o nome do aplicativo foi retirado dos quadros de análise de software nas próximas seções deste capítulo.

6.3.8 MicroERP

O MicroERP é um sistema desenvolvido em Java que oferece inclusive suporte para aplicações *e-commerce*. Dentre os módulos que o sistema diz oferecer estão: pedidos, financeiro, contas a pagar e a receber, recursos humanos e controle de estoque.

Infelizmente não foi possível avaliar o MicroERP da forma como os outros softwares foram avaliados, pois, apesar da instalação ser realizada conforme a documentação estabelecia, no momento da inicialização do sistema, quando era solicitada a senha de acesso, esta não estava presente em nenhum lugar da documentação. Foi tentado contato por *e-mail* com os desenvolvedores relatando o ocorrido, e embora tenho sido recebida uma resposta com uma senha para o sistema, a mesma não funcionou. Outras tentativas de contato não obtiveram retorno.

Em função do ocorrido, o nome do MicroERP será omitido dos quadros comparativos dos sistemas.

6.3.9 openXpertya

O openXpertya é um *software* escrito em Java que possibilita utilização em três camadas. O aplicativo teve sua origem baseado nas características do sistema Compiere. Infelizmente o openXpertya apresenta uma limitação do

ponto de vista da realização desta pesquisa: o banco de dados utilizado pelo aplicativo é o Oracle, embora exista, em projeto, a intenção de tornar possível a utilização do sistema em outros bancos de dados com características de software livre, esta funcionalidade ainda não está disponível. Também não existem outras versões portadas para outros bancos de dados, conforme ocorreu no caso do Compiere.

Por tratar-se o Oracle de um banco de dados proprietário e comercial impõe ao openXpertya uma característica que faz necessária a aquisição de licenças comerciais para utilização da base de dados, o que foge ao padrão dos outros sistemas instalados para este trabalho. Em função deste aspecto, as informações sobre o openXpertya aqui apresentadas foram coletadas diretamente no *site* do desenvolvedor.

A licença utilizada pelo aplicativo é uma licença própria, chamada de LPO (Licença Pública do openXpertya) que não é viral e está baseada na licença dos aplicativos da empresa Sun e na licença MPL.

O desenvolver promete características para gerenciamento de várias empresas, controle exposto em diferentes tipos de moeda, diferentes tipos de contabilidade e diferentes métodos de custo, além de uma interface fácil de utilizar e funcionalidades de análise de dados por meio de ferramenta OLAP (Online Analytical Processing). Também são oferecidas possibilidades de integração e exportação em arquivos nos formatos Excel, HTML, XML, Texto, PDF, PS e Word.

Por motivo de não ter sido instalado, este aplicativo não aparece nos quadros comparativos das próximas seções.

6.3.10 Oratio

Projetado para utilização com recursos do navegador *web*, o Oratio não requer hardware sofisticado ou que não se encontre no ambiente das pequenas empresas. A instalação foi considerada como sendo de nível médio de dificuldade pois exige a descompactação de arquivos, além da instalação das bibliotecas que permitam o uso da linguagem Perl para acesso ao banco de dados e configurações adicionais no servidor de páginas *web*.

O Oratio utiliza o Postgresql para gerenciar os dados armazenados pelo sistema e não possibilita que sejam gerenciadas mais de uma empresa.

Os recursos de emissão de relatório incluem pedidos de venda, entregas, vendas realizadas, compras, produção, níveis de estoque, livro caixa e relatórios fiscais. Os relatórios são apresentados na tela e podem ser impressos pelos recursos de impressão do navegador *web*. Também é possível exportar os relatórios em formato de arquivo PDF, CVS ou HTML, mas não é possível gerar gráficos a partir do sistema.

O aplicativo dispõe de um recurso para que a geração de cópia de segurança, que pode inclusive ser enviada para uma conta de *e-mail* integrando assim uma ferramenta de comunicação ao sistema.

É necessária uma senha para acesso ao sistema e podem ser criados diferentes usuários e grupos com diferentes tipos de permissão de acesso ao sistema. No quadro 17, pode-se verificar informações sobre o aplicativo Oratio.

Nome: Oratio	
Web site: http://oratio-project.org	
Sistema operacional: Windows, POSIX	Tipo de licença: GPL
Nº inscritos grupo discussão: 133	
Indicação hardware: equipamento que suporte sistema operacional escolhido	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Média	
Software adicional: Postgresql, Servidor <i>Web</i> (Apache), Active Perl, Bibliotecas DBI e DBD-Pg	
Banco de dados nativo: Postgresql	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Sim
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Sim	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim (grupo de discussão e empresa prestadora de serviço, mas nenhuma no Brasil)

Quadro 17 - Informações Oratio

Com uma interface que procura ser clara e de fácil utilização, o Oratio possui também documentação detalhada sobre instalação e utilização, além de um número de 133 participantes em lista de discussão sobre o sistema.

O principal responsável pelo desenvolvimento do sistema é uma empresa (Centauri) que também oferece serviços de manutenção, integração e hospedagem, mas o serviço ainda não está disponível para o Brasil.

Uma limitação deste aplicativo para uso na pequenas empresas brasileiras é ainda não estar disponível no idioma Português.

6.3.11 Sequóia ERP

Desenvolvido utilizando a linguagem Java, o Sequóia ERP é um sistema baseado em um conjunto de aplicativos chamado OfficeBiz que pode ser utilizado para a construção de sistemas. O OfficeBiz funciona como um modelo sob o qual podem ser feitas alterações para adaptações necessárias.

O *hardware* indicado pelo desenvolvedor é um computador que suporte os sistemas operacionais Windows NT/2000/XP ou POSIX.

O suporte para instalação e configuração conta com uma lista de discussão com poucos participantes (29 membros) e empresas que oferecem serviços, mas nenhuma delas atende no Brasil. A solução é fazer uso da documentação disponível tanto sobre a instalação como forma de utilização que se encontra acessível por meio de páginas *web*.

O Sequóia ERP faz uso de um interpretador de código Java (JSP – Java *Server Pages*) e de páginas html para construção de seu ambiente de funcionamento.

A instalação consiste da descompactação de arquivos, depois basta executar um arquivo para que este faça funcionar o interpretador de páginas possibilitando que o sistema seja utilizado com o apoio de um navegador *web*. Isto fez com que o nível de dificuldade de instalação fosse considerado baixo.

A interface do *software* é agradável e de fácil compreensão contando inclusive com um mecanismo para minimizar erros de preenchimento dos campos, como, por exemplo, calendários em campos de data.

Com relação aos relatórios, estes podem ser visualizados na tela ou enviados à impressora, além da possibilidade de gerar relatórios em arquivos de formato PDF, mas não é possível gerar gráficos.

No quadro 18, são mostradas informações sobre o aplicativo Sequóia ERP.

Nome: Sequóia ERP	
Web site: http://sequoiaerp.org	
Sistema operacional: Todos	Tipo de licença: MIT
Nº inscritos grupo discussão: 29 pessoas	
Indicação hardware: equipamento que suporte instalação dos sistemas operacionais Windows NT/2000/XP ou POSIX	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Baixo	
Software adicional: Mysql	
Banco de dados nativo: MySQL	
Idioma – disponível em português: Sim	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Não
Documentação disponível : Sim	Suporte: Sim (lista <i>e-mail</i> e empresas prestadoras de serviços, mas nenhuma no Brasil).

Quadro 18 - Informações Sequóia ERP

O sistema não possui características de integração com outros aplicativos nem ferramentas para envio de mensagens integradas.

O acesso ao aplicativo faz uso de senhas e é possível criar diferentes níveis de permissão para diferentes tipos de usuário, mas a política de *backup* deve ser implementada de forma externa ao aplicativo.

6.3.12 Tiny ERP

Com possibilidade de utilização em todos os sistemas operacionais, o Tiny ERP é um aplicativo desenvolvido utilizando uma linguagem chamada Python e não tem nenhum requisito especial de *hardware* para sua instalação.

O sistema foi desenvolvido utilizando as características do modelo cliente-servidor não só para o banco de dados, mas também para o funcionamento do próprio sistema. Para seu correto funcionamento, é preciso instalar as duas partes do sistema: o cliente e o servidor. Embora seja necessário realizar estas duas instalações, o processo é feito de forma bem simples com a ajuda de dois instaladores para auxiliar no processo de instalação e configuração do aplicativo: um instalador para o servidor e outro para o cliente, tornando a instalação bastante simples.

Quanto ao banco de dados, o Tiny ERP funciona com o Postgresql e não indica a possibilidade de utilização de outro gerenciador de banco de dados. A importação dos dados que a empresa por acaso possa já ter antes da instalação do sistema pode ser feita com o uso de arquivos *.sql.

O Tiny ERP permite que sejam gerenciadas mais de uma empresa ao mesmo tempo, mas não no mesmo banco de dados, nem por meio do mesmo servidor da aplicação. Para gerenciar uma segunda empresa é necessário configurar um outro programa servidor e direcioná-lo para um endereço (porta) diferente daquela onde o primeiro programa servidor está funcionando. Isto limita bastante a possibilidade de utilizar o sistema para gerenciar mais de uma empresa, além de impor dificuldades ao processo de configuração do segundo servidor.

Relatórios referentes a parceiros, produtos, vendas, pedidos, estoque, projetos, finanças e contabilidade estão presentes no sistema. Os relatórios podem ser vistos na tela ou enviados à impressora ou ainda transformados em arquivos no formato PDF ou formato de texto.

Não existe a possibilidade de se criar novos relatórios ou alterar os já existente por meio de uma ferramenta oferecida pelo aplicativo. A exemplo das alterações em outro aplicativos, isso só pode ser feito mediante a alteração do código fonte do programa.

O Tiny ERP não possui recursos para integração com outro software, mas possui uma ferramenta para envio de mensagens a outros usuários que é parte integrante do sistema.

É possível criar diferentes níveis de acesso por parte do usuário ao sistema e diferentes regras de acesso para da usuário. O aplicativo não oferece recursos internos para *backup*, ficando a política de *backup* por conta do administrador do sistema.

No quadro 19, encontram-se disponíveis mais informações sobre o Tiny ERP.

Nome: Tiny ERP	
Web site: http://tiny.be ou http://tinyerp.com	
Sistema operacional: Todos	Tipo de licença: GPL
Nº inscritos grupo discussão: não possui	
Indicação hardware: equipamento que suporte o sistema operacional escolhido	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Sim	Opção de desinstalação: Sim
Dificuldade de instalação: Baixa	
Software adicional: Postgresql	
Banco de dados nativo: Postgresql	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Sim	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Sim
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim (via <i>e-mail</i>)

Quadro 19 - Informações Tiny ERP

A interface de trabalho do Tiny ERP é bastante agradável com botões, objetos e mensagens claras e de fácil interpretação. Existe um mecanismo de uso de cores que informa quais são os campos que obrigatoriamente devem ser preenchidos nas telas do sistema.

O aplicativo oferece documentação de instalação e utilização disponibilizando esta documentação via internet (páginas *web*). Embora não haja um grupo de discussão cadastrado para a troca de informações e possíveis dúvidas ou dificuldades, é possível enviar *e-mail* para os desenvolvedores.

6.3.13 WebERP

Este aplicativo teve início de desenvolvimento em 2001. Por ser desenvolvido utilizando a linguagem PHP o WebERP possui a versatilidade de funcionar em qualquer navegador *web*.

O WebERP pode funcionar em qualquer computador onde possa ser instalada a plataforma AMP. Os desenvolvedores informam a possibilidade de uso em qualquer sistema operacional do tipo POSIX (Linux, Unix) ou Windows.

Não foram encontradas indicações específica do *hardware* necessário e a instalação se deu sem problemas no equipamento utilizado para testes. A licença utilizada é a GPL.

O WebERP não possui um instalador e a instalação é feita por meio da descompactação dos arquivos do aplicativo no local designado pelo usuário. Em seguida, é necessário criar um banco de dados e configurar determinados arquivos.

Utiliza como gerenciador de banco de dados nativo o MySQL, mas pode também ser utilizado com Postgresql. A importação de dados de possíveis bases já existentes na empresa pode ser feita por meio do utilitário chamado phpMyAdmin ou externamente por meio da transformação dos dados em formato de arquivos *.sql.

As alterações necessárias em campos da tabela do banco de dados ou alterações na interface do programa só é possível por meio da alteração do próprio código fonte do aplicativo e da manipulação direta das propriedades das tabelas.

Os relatórios gerados podem ser visualizados na tela do computador, transformados em arquivos de formato PDF (*Portable Document Format*) ou impressos a partir do menu de impressão do navegador *web*. A quantidade de relatório oferecidas pelo WebERP é bastante satisfatória, conforme pode ser verificado na listagem apresentada a seguir:

- Pedidos;
- Listas de preços ;
- Posição dos Pedidos ;
- Pedidos Faturados;
- Diferenças na entrega ;
- Transações de clientes;
- Local de alocação;
- Análise de vendas;
- Saldos/dívidas pendentes;
- Transações;
- Lista de depósitos;
- Saldos Devedor Meses anteriores;
- Clientes por área/vendedor;
- Fornecedores;
- Pagamentos;

- Pendências;
- Saldo Credor em meses anteriores;
- Movimentações de item de inventário;
- Estado do item de inventário;
- Uso do item no inventário ;
- Avaliação do inventário;
- Planejamento de inventário;
- Planilhas de verificação de estoque no inventário;
- Criar arquivo CSV de quantidades no inventário;
- Comparar contagens com dados de verificação de estoque;
- Todas as movimentações de estoque por localidade/data;
- Estado do inventário por localidade/categoria;
- Quantidades no estoque históricas por localidade/categoria;
- Estoques Negativos;
- Faturas de material;
- Onde é usado um item de estoque;
- Balancete Comercial;
- Contas;
- Conciliação bancária;
- Pagamentos via cheque;
- Lucros e prejuízos;
- Balanço geral;
- Relatórios de Impostos;

O WebERP possui ainda a possibilidade de gerar gráficos de colunas com os dados de vendas.

Não estão presentes neste aplicativo características de integração direta com outros softwares como planilhas eletrônicas ou ferramentas de *e-mail*.

Quanto à segurança, é possível estabelecer diferentes tipos de permissão para diferentes tipos de acesso, como por exemplo: somente para consultas, ou acesso limitado à determinada área do aplicativo (estoque, compras, etc).

Já a política de *backup* fica por conta do usuário para ser realizada de forma externa ao software que não oferece um recurso interno para a execução da cópia de segurança.

Os aspectos de usabilidade foram levados em consideração e o aplicativo apresenta boa documentação tanto para a instalação quanto para utilização. As informações para utilização estão disponíveis diretamente no sistema, não sendo necessário a busca em arquivos ou *site*. As mensagens e os objetos que constituem a interface são apresentados de forma compreensível e existe suporte para o português do Brasil.

Além de uma comunidade de usuários com 208 membros, oferecendo ajuda e suporte por meio de listas de discussão, existe para outros países a possibilidade de contratar empresas que oferecem suporte comercial para instalação, configuração e adaptação. No Brasil não existe nenhum suporte oficial.

No Quadro 20 encontram-se disponíveis mais informações sobre o webERP.

Nome: webERP	
Web site: http://weberp.org	
Sistema operacional: Windos, POSIX	Tipo de licença: GPL
Nº inscritos grupo discussão: 208	
Indicação hardware: 512 Mb de memória RAM para o servidor	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Média	
Software adicional: plataforma AMP	
Banco de dados nativo: MySQL	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Sim
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fácies e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim (Grupos de discussão e empresas prestadoras de serviço, mas nenhuma no Brasil).

Quadro 20 – Informações webERP

6.3.14 University

Desenvolvido para atender o processo de negócio de empresas que atuam na área de educação, o University foi construído utilizando a linguagem Java.

Assim como todos os outros sistemas analisados, o hardware que o University requer está presente no ambiente de uma pequena empresa.

A instalação têm um baixo grau de dificuldade, bastando descompactar os arquivos e executar um arquivo chamado “startup.bat”.

O MySQL é o servidor de banco de dados com o qual o University funciona, mas pode ser também instalado com o Oracle. A alteração de campos de tabelas e interface com usuário só pode ser feita com alteração do código fonte.

Existe pouca possibilidade de utilização de relatórios, com destaque para o pagamento de boletos e alunos. Embora o University seja um projeto para atender as atividades de uma universidade ou outros tipos de escola, o sistema ainda não está completo.

O aplicativo requer senha para acesso ao sistema, e possibilita a criação de diferentes tipos de usuário: professores, alunos e administradores. Detalhes do aplicativo são apresentados no Quadro 21.

Nome:University	
Web site: http://www.red1.org/university	
Sistema operacional:Todos	Tipo de licença: Apache Licence
Nº inscritos grupo discussão: não tem	
Indicação hardware: não tem	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Baixa	
Software adicional: MySQL	
Banco de dados nativo: MySQL	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Não	Gerador de gráficos: Não
Possibilita integração outro software: Não	Ferramentas comunicação: Não
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fácies e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim (Grupos de discussão e empresas prestadoras de serviço, mas nenhuma no Brasil, mas oferece serviço on-line).

Quadro 21 – Informações University

A interface com usuário é bastante intuitiva e disponibiliza documentação por meio da *internet*. O desenvolvedor oferece suporte para instalação com a realização de contratos.

6.3.15 Value

O Value já utilizou diferentes linguagens na construção de seu código, mas a versão testada neste trabalho foi a que utiliza PHP. Existem dois grupos de discussão aos quais o usuário pode recorrer para tirar suas dúvidas, porém a instalação do Value obedece aos requisitos da instalação dos outros programas desenvolvidos em PHP.

O aplicativo não possui instalador ou desinstalador e requer que seja instalada a plataforma AMP para que possa funcionar corretamente.

A importação de dados que possam existir só é possível mediante o uso de arquivos do tipo *.sql. As possíveis alterações em tabela e interface só é possível com alteração do código fonte do sistema.

Os relatórios oferecidos pelo sistema podem ser exportados em arquivos nos formatos Excel, Word, CSV ou XML. É possível também gerar gráficos de colunas, pizza e barras com as informações.

O quadro 22 apresenta mais informações sobre o aplicativo Value.

Nome: Value	
Web site: http://value.sourceforge.net	
Sistema operacional: Todos	Tipo de licença: LGPL
Nº inscritos grupo discussão: não tem	
Indicação hardware: não tem	
Hardware presente no ambiente pequenas empresas: Sim	
Possui instalador: Não	Opção de desinstalação: Não
Dificuldade de instalação: Baixa	
Software adicional: LAMP	
Banco de dados nativo: MySQL	
Idioma – disponível em português: Não	
Gerencia mais de uma empresa: Não	Possui relatórios: Sim
Exporta relatórios em arquivos: Sim	Gerador de gráficos: Sim
Possibilita integração outro software: Sim	Ferramentas comunicação: Não
Senha para acesso: Sim	Diferentes níveis de usuário: Sim
Backup interno: Não	Interface natural: Sim
Expressões habituais usuário: Sim	Objetos e ações fáceis e visíveis: Sim
Indicações de resultados e controles: Sim	Mecanismo detecção de erros: Sim
Mensagens clara de erros: Sim	Solução dúvidas utilização: Sim
Documentação disponível : Sim	Suporte: Sim (Grupos de discussão)

Quadro 22 – Informações Value

É necessário uma senha para acesso ao sistema e existe possibilidade de permissões diferentes de acesso para diferentes tipos de usuários.

A usabilidade do aplicativo é boa, mas a documentação somente é oferecida por meio de acesso à *internet*.

6.4 Viabilidade de instalação

Esta seção apresenta os resultados das questões referentes à primeira variável da pesquisa: a viabilidade de instalação dos aplicativos.

6.4.1 Nível de dificuldade de instalação

Com respeito à dificuldade de instalação, a maioria dos aplicativos apresentou dificuldade média ou baixa. Apenas o Compiere obteve a classificação alta. Os aplicativos que se utilizam da plataforma AMP, embora tenham sido classificados com dificuldade média, podem ser instalados com a ajuda de profissionais que desenvolvem página dinâmicas para ambiente *web*, pois são profissionais que estão acostumados com esta forma de instalação e configuração.

A definição do nível de dificuldade foi feita com base no número de atividades necessárias para que a instalação do aplicativo fosse concretizada, bem como no nível de conhecimento necessário para que estas atividades fossem realizadas. Assim uma dificuldade de instalação média, por exemplo, implica em instalar programas adicionais (banco de dados, servidor *web*, interpretador de código) por meio de utilitários que automatizam esse processo de instalação. Um baixo nível de dificuldade implica em ter um utilitário que se encarregue de todo o processo de instalação, sem que o usuário precise realizar instalações de programas adicionais e um alto nível de dificuldade indica que o usuário além de instalar programas adicionais vai precisar configurar detalhes da instalação em arquivos ou variáveis de sistema, necessitando um maior conhecimento computacional.

O Quadro 23 apresenta os aplicativos e seus níveis de dificuldade de instalação.

Aplicativos	Dificuldade de Instalação		
	Alta	Média	Baixa
Compiere	X		
Evaristo		X	
EzyBiz		X	
Factura LUX			X
Freedom ERP			X
Oratio		X	
Sequoia ERP			X
Tiny ERP			X
University ERP System			X
Value		X	
WebERP		X	

Quadro 23 – Nível de dificuldade de instalação

6.4.2 Necessidade de software adicional

Conforme apresentado no Quadro 24, a maioria dos aplicativos é dependente da instalação e configuração de outros aplicativos que dão suporte ao seu correto funcionamento. A única exceção foi o *software* FacturaLux que instala todos os itens necessários a partir de seu próprio programa de instalação.

A necessidade da instalação de software adicional é evidenciada pelas linguagens e banco de dados dos quais os aplicativos fazem uso. A linguagem Java necessita de um interpretador instalado para cada tipo de sistema operacional no qual se deseja utilizar o aplicativo, assim como o PHP e Python. Isto é interessante porque dá ao aplicativo a possibilidade de funcionar em diversos sistemas operacionais se adaptando ao conhecimento e à disponibilidade do usuário.

As necessidades de software adicional podem ser vistas no Quadro 24.

Aplicativos	Software adicional
Compiere	J2SE (Java to Standard Edition) JDK (Java Development Kit) Postgresql
Evaristo	J2SE (Java to Standard Edition) Runtime Environment Postgresql
EzyBiz	Sevidor <i>Web</i> (Apache) PHP MySQL
Factura LUX	Nenhum
Freedom ERP	J2SE (Java to Standard Edition) Runtime Environment Firebird
Oratio	Active Perl - DBI-1.4.7 e DBD-pg 1.22 Servidor <i>Web</i> (Apache) SGBD Postgresql
Sequoia ERP	MySQL
Tiny ERP	Postgresql
University ERP System	MySQL
Value	Sevidor <i>Web</i> (Apache) PHP MySQL
WebERP	Sevidor <i>Web</i> (Apache) PHP MySQL

Quadro 24 – Software adicional para cada aplicativo instalado

6.4.3 Recursos de instalação

Em função das próprias características das linguagens de desenvolvimento utilizadas e da necessidade de instalação de software adicional, apenas dois aplicativos: FacturaLux e Tiny ERP apresentaram um software de instalação e desinstalação. O Quadro 25 apresenta os aplicativos e relaciona a existência de um programa instalador.

Aplicativo	Possui programa para:	
	Instalar	Desinstalar
Compiere	Não	Não
Evaristo	Não	Não
EzyBiz	Não	Não
Factura LUX	Sim	Sim
Freedom ERP	Não	Não
Oratio	Não	Não
Sequoia ERP	Não	Não
Tiny ERP	Sim	Sim
University ERP System	Não	Não
Value	Não	Não
WebERP	Não	Não

Quadro 25 – Presença de programa de instalação / desinstalação

6.5 Funcionalidades

Esta seção trata da segunda variável de pesquisa: as funcionalidades presentes nos sistemas integrados de gestão analisados.

6.5.1 Acesso a banco de dados

Os aplicativos instalados mostraram-se flexíveis à escolha de diferentes gerenciadores de banco de dados. A maioria deles pode fazer uso de mais de um tipo de gerenciador (vide Quadro 26). Como destaque, apresentam-se o MySQL e o Postgresql que são gerenciadores de banco de dados desenvolvidos também sobre os moldes do software livre, levando inclusive ao desenvolvimento de diferentes versões do aplicativo Compiere para funcionar utilizando um banco de dados que não fosse proprietário.

O fato da possibilidade de escolha no tipo de banco de dados a ser utilizado é mais um ponto a favor da flexibilidade para atender aos conhecimentos e necessidades dos usuários com respeito à integração de informações.

Aplicativo	Acesso a banco de dados	
	Nativo	Outros
Compiere *	Oracle, Postgresl, Fyracle	Não
Evaristo	Postgresql	Oracle, DB2, SQL Server
EzyBiz	MySQL	Não
Factura LUX	SQLite	Postgresql
Freedom ERP	Firebird	Não
Oratio	Postgresql	Não
Sequoia ERP	MySQL	Postgresql
Tiny ERP	Postgresql	Não
University ERP System	MySQL	Oracle
Value	MySQL	Não
WebERP	MySQL	Postgresql

* Dependendo da versão instalada
 Quadro 26 – Acesso a banco de dados

6.5.2 Adequação de dados à empresa

Com exceção do Compiere, todos os outros aplicativos não possuem ferramentas integradas que permitam a criação de novos campos às tabelas do banco de dados. Esta ferramenta seria interessante para facilitar a adequação dos dados às necessidades específicas da empresa. Como o código fonte está disponível, é possível proceder as alterações necessárias tanto na base de dados quanto na interface do programa, porém a realização das modificações desta forma exige conhecimento técnico e demanda um tempo maior para a compreensão do código e interpretação da estrutura das tabelas criadas.

Outro detalhe com relação à adequação de dados diz respeito à possibilidade do sistema ser capaz de gerenciar informações referentes a mais de uma empresa, no caso da existência de uma filial, por exemplo. A maioria dos sistemas não incorpora esta capacidade, ou o faz parcialmente, possibilitando o gerenciamento de mais de uma empresa, sem contudo possibilitar relatórios que integrem ou comparem as informações de diferentes empresas. O quadro 27 sintetiza estas informações.

Aplicativo	Criar ou alterar tabela de dados	Ajustes na interface	Gerencia mais de uma empresa
Compiere	Ferramenta integrada	Ferramenta integrada	Sim
Evaristo	Diretamente na tabela	Alterando código fonte	Sim
EzyBiz	Diretamente na tabela	Alterando código fonte	Sim
Factura LUX	Diretamente na tabela	Alterando código fonte	Sim
Freedom ERP	Diretamente na tabela	Alterando código fonte	Sim
Oratio	Diretamente na tabela	Alterando código fonte	Não
Sequoia ERP	Diretamente na tabela	Alterando código fonte	Não
Tiny ERP	Diretamente na tabela	Alterando código fonte	Sim
University ERP System	Diretamente na tabela	Alterando código fonte	Não
Value	Diretamente na tabela	Alterando código fonte	Não
WebERP	Diretamente na tabela	Alterando código fonte	Não

Quadro 27 – Adequação de dados

6.5.3 Módulos

Os sistemas instalados possuem diferentes disponibilidades de módulos, o que pode ajudar a determinar qual aplicativo possui melhor adequação à empresa, dependendo de sua atividade, interesse e partes funcionais do negócio onde a empresa sente a maior necessidade de informatização.

O quadro 28 apresenta informações sobre quais módulos estão presentes nos sistemas. Os módulos são apresentados em sua principal descrição, normalmente por áreas funcionais, mas para cada um deles é possível encontrar opções, como por exemplo, no módulo de vendas, uma opção de realização de orçamento ou pedido.

Também foi mantida a descrição encontrada no aplicativo, por exemplo, o que é financeiro em um sistema pode estar representado por contas a pagar e contas a receber em outro.

Aplicativos	Módulos
Compiere	regras – parceiros, vendas, compras, materiais, projetos, performance
Evaristo	entidades, produtos, documentos, listagens
EzyBiz	vendas, projetos, financeiro, recursos humanos, compras, estoque
Factura LUX	estoque, financeiro, livro-caixa
Freedom ERP	financeiro, produtos, ponto de venda, vendas, atendimento e clientes
Oratio	projetos, compras, vendas, estoque, produção, financeiro
Sequoia ERP	contabilidade, manufatura, marketing, pedidos, peças, CMS
Tiny ERP	parceiros, financeiro, vendas, compras, estoque, produção e projetos
University ERP System	cursos, alunos, disciplinas, financeiro
Value	Contabilidade, contas a pagar, contas receber, estoque, vendas, manufatura
WebERP	Compras, financeiro, produção, estoque, cobrança

Quadro 28 - Módulos

Não foram encontrados padrões para os nomes dos módulos e o conteúdo à que cada um se refere está diretamente ligado à forma de funcionamento do sistema. O aplicativo Evaristo, por exemplo, possui um módulo chamado “Entidade” em que se encontram as contas da empresa, seus clientes e fornecedores, enquanto outros aplicativos como o Freedom ERP têm um módulo exclusivo para o tratamento das informações de clientes.

6.5.4 Relatórios e gráficos

Todos aplicativos ofereceram recursos para geração de relatórios atendendo as áreas às quais o aplicativo atua. Alguns dos aplicativos possuem um número maior de relatórios. No quadro 29, são apresentados os principais tópicos a que se referem os relatórios, lembrando que normalmente cada um dos tipos apresenta especificações como, por exemplo, um relatório de pedidos por data ou por cliente.

Aplicativos	Tipos de relatório
Compiere	Praticamente todas as consultas podem ser impressas
Evaristo	Compras, vendas, produtos, clientes
EzyBiz	Pedido, contas, estoque, compras, folha de pagamento, livro caixa, atividade de projetos
Factura LUX	Livro caixa, faturamento, vendas, compras, inventário
Freedom ERP	Financeiro, produtos, ponto de venda, vendas, atendimento a clientes
Oratio	Pedidos, vendas, compras, produção, estoques, livro caixa, relatórios fiscais
Sequoia ERP	Pedidos, produtos, clientes
Tiny ERP	Parceiros, produtos, pedidos, estoque, projetos, finanças, contabilidade, vendas
University ERP System	Boletos e recebimento de boletos
Value	Contabilidade, contas pagar, receber, estoque, vendas e produção
WebERP	Pedidos, entregas, vendas, fornecedores, pagamentos, estoque, produtos, contabilidade,

Quadro 29 – Tipos de relatórios

Todos os aplicativos disponibilizam os relatórios na tela para que, posteriormente, possam ser enviados à impressora, seja por recurso do próprio *software*, seja pelo uso dos recursos de impressão do navegador *web*. Alguns aplicativos também oferecem a possibilidade de gerar os relatórios em formato de arquivos. Isto é particularmente importante caso a empresa necessite arquivar relatórios, informações sobre um determinado período ou fato, podendo manter um arquivo digital destas informações. Gerar relatórios em arquivos também é útil caso seja necessário o envio de relatório para outra empresa ou fornecedor, facilitando o uso de ferramentas como, por exemplo, o *e-mail* para que o relatório seja enviado. O formato de arquivo mais comum no qual os relatórios são gerados é o PDF (Portable Document Format), mas também foram encontrados aplicativos que utilizam o formato TXT (texto simples), CSV (dados separados por vírgula), XML (eXtesible Markup Language) e HTML (Hiper Text Markup Language).

Além de relatórios em arquivos, alguns aplicativos apresentam também a possibilidade de gerar gráficos com as informações. Este recurso é particularmente importante para a visualização e significado dos dados que a empresa dispõe, facilitando assim o processo de tomada de decisão.

O quadro 30 apresenta as informações sobre estes recursos de relatório.

Aplicativo	Relatório na tela e na impressora	Relatório em arquivo - formato	Gera gráficos
Compiere	Sim	Sim – PDF, XML	Sim
Evaristo	Sim	Não	Não
EzyBiz	Sim	Não	Sim
Factura LUX	Sim	Não	Não
Freedom ERP	Sim	Sim – PDF, TXT	Não
Oratio	Sim	Sim – PDF, CVS, HTML	Não
Sequoia ERP	Sim	Sim – PDF	Não
Tiny ERP	Sim	Sim – PDF, TXT	Sim
University ERP System	Sim	Não	Não
Value	Sim	XLS, DOC, CVS, XML	Sim
WebERP	Sim	Sim – PDF	Sim

Quadro 30 – Características dos relatórios

Com exceção do Compiere, as alterações que se tornarem necessárias para adequação dos relatórios às necessidades do usuário ou novos relatórios só são possíveis mediante a manipulação do código fonte do aplicativo.

6.5.5 Integração com outros softwares

Nenhum dos sistemas integrados incorpora ferramentas de integração com outro *software*, embora haja projetos de fazê-lo. O Compiere, por exemplo, tem uma proposta de implementação de leitura e exportação de arquivos no formato de planilha eletrônica do tipo Excell. Assim, o mais próximo que foi identificado com relação à esta característica é o fato de alguns aplicativos exportarem seus relatórios em formato de arquivos que podem ser usados em outros programas, como por exemplo, o Value que já exporta arquivos em formato do aplicativo Excel.

Com relação à recursos de comunicação presentes nos aplicativos, o Tiny Erp possui uma ferramenta para envio de mensagens aos usuários do sistema e o Compiere possibilita o envio de *e-mails*.

6.5.6 Segurança

No aspecto segurança de acesso ao sistema, a maioria dos aplicativos apresenta recursos para o uso de senhas de acesso e em vários deles é possível criar diferentes regras para diferentes tipos de usuários. Apenas o Evaristo não apresentou esta característica.

O quadro 31 relaciona os aplicativos e o aspecto segurança de acesso e cópias de segurança.

Aplicativo	Senha para acesso	Diferentes níveis de permissão	Ferramenta integrada de <i>backup</i>
Compiere	Sim	Sim	Não
Evaristo	Não	Não	Sim
EzyBiz	Sim	Sim	Não
Factura LUX	Sim*	Sim*	Não
Freedom ERP	Sim	Sim	Não
Oratio	Sim	Sim	Sim
Sequoia ERP	Sim	Sim	Não
Tiny ERP	Sim	Sim	Não
University ERP System	Sim	Sim	Não
Value	Sim	Sim	Não
WebERP	Sim	Sim	Não

* implementado por módulo pago
Quadro 31 – Segurança de acesso

No que diz respeito à cópias de segurança (*backup*), os aplicativos Evaristo e Oratio foram os únicos a disponibilizam este recurso de forma integrada ao sistema. A maioria dos aplicativos não apresentou uma ferramenta que permitisse a realização do *backup* a partir do próprio *software*. Esta decisão parece seguir o conceito de utilização da filosofia do próprio Unix em que cada aplicativo deve ser preocupar em fazer bem exatamente aquilo para o que foi criado. Assim, existindo ferramentas disponíveis no sistema operacional para o agendamento de tarefas e recursos oferecidos nos gerenciadores de banco de dados para gerar arquivos de cópia de segurança, bastaria combinar essas possibilidades e obter as cópias nos momentos e em quantas vezes o usuário julgar necessário.

6.6 Usabilidade

O método da Heurística utilizado para a avaliação de usabilidade mostrou que os aplicativos de sistemas integrados de gestão desenvolvidos sob o modelo de *software* livre têm apresentado boa usabilidade. Poucos são os casos em que não houve preocupação em informar ao usuário com mensagens claras ou faltou a presença de mecanismos de auxílio à detecção de erros como, por exemplo, campos de dados referentes a datas com verificação de validade ou presença de calendário para ajuda no preenchimento.

A construção de uma interface com objetos bastante visíveis e que representam de forma clara e intuitiva as ações às quais estão associados também está presente nos aplicativos. As falhas neste aspecto só foram detectados no Ezybiz.

No Quadro 32 estão sintetizadas informações a respeito do aspecto usabilidade.

Aplicativo	Interface natural	Expressões habituais usuário	Objetos e ações fáceis e visíveis	Indicações resultados e controles	Mecanismo detecção erros	Mensagens claras de erros	Mecanismo solução dúvidas	Documentação
Compiere	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Evaristo	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
EzyBiz	Sim	Sim	Não	Sim	Sim	Não	Não	Sim
Factura LUX	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Freedom ERP	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não
Oratio	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Sequoia ERP	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Tiny ERP	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
University ERP System	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Value	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
WebERP	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim

Quadro 32 – Aspectos de usabilidade

Outro ponto importante a ser verificado foi a disponibilidade do idioma Português (Brasil) para o sistema. Este aspecto aumenta a possibilidade de uso do aplicativo. Vale lembrar que alguns dos sistemas citavam a capacidade de utilização em Português, mas, no momento da instalação, esta opção não podia ser habilitada, ou era solicitada durante a instalação mas o aplicativo não

utilizava o idioma escolhido depois de concluída a instalação. Foi o que aconteceu, por exemplo, com versões testadas do Compiere, onde só foi encontrada a possibilidade de uso em Português na versão desenvolvida pela empresa brasileira. Uma dica interessante é a verificação da lista de melhorias e implementações a ser feitas no aplicativo (do inglês “*To do*”) que se encontra disponível no local onde o projeto de software está armazenado. Foi ali que foi descoberto que embora a opção de vários idiomas estivesse disponível na configuração do programa Compiere, o Português ainda não havia sido implementado na versão original do sistema.

O Quadro 33 representa os idiomas em que os aplicativos se mostraram efetivamente disponíveis, bem como a forma de encontrar suporte à instalação, adequação e desenvolvimento.

Aplicativo	Idiomas	Suporte
Compiere	alemão, chinês simplificado, chinês tradicional, dinamarquês, espanhol, francês, indonésio, inglês, italiano, malaio, romeno, russo, suíço e português*	- Suporte comercial em mais de 60 empresas no mundo (inclusive no Brasil). - Programa de treinamento e certificação (em inglês).
Evaristo	Português (Portugal)	Suporte via <i>e-mail</i> e comercial
EzyBiz	Inglês	Não há suporte
Factura LUX	Espanhol, português e alemão (mas somente espanhol funcionou)	Serviços oferecidos pelo desenvolvedor, mas não disponível no Brasil
Freedom ERP	Português (Brasil)	- via <i>e-mail</i> - Suporte comercial disponível no Brasil
Oratio	espanhol, francês, grego, inglês e italiano	Serviços oferecidos pelo desenvolvedor, mas não disponível no Brasil
Sequoia ERP		- Lista de discussão. - Suporte comercial não disponível no Brasil
Tiny ERP	alemão, espanhol, francês e inglês	via <i>e-mail</i>
University ERP System	Inglês	Serviços oferecidos pelo desenvolvedor, com contrato ou on- line
Value	Inglês	Não – somente documentação
WebERP	espanhol, inglês, polonês, português (Brasil) e russo	- Lista de discussão. - Suporte comercial não disponível no Brasil

* somente na versão produzida no Brasil
Quadro 33 – Idiomas e formas de suporte

O quadro 34 representa um comparativo das principais características dos aplicativos.

Aplicativo	Dificuldade instalação	Relatório em arquivo	Gera gráficos	Senha para acesso	Diferentes níveis de permissão	Ferramenta integrada de <i>backup</i>	Disponível em português	Interface natural	Expressões habituais usuário	Objetos e ações fáceis e visíveis	Indicações, resultados e controles	Mecanismo detecção erros	Mensagens claras de erros	Mecanismo solução dúvidas	Suporte disponível no Brasil	Documentação
Compiere	alta	Sim	Sim	Sim	Sim	Não	Sim*	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Evaristo	média	Não	Não	Não	Não	Sim	Não	Sim	Sim	Sim	Sim	Não	Sim	Sim	Não*	Sim
EzyBiz	média	Não	Sim	Sim	Sim	Não	Não	Sim	Sim	Não	Sim	Sim	Não	Não	Não	Sim
Factura LUX	baixa	Não	Não	Sim*	Sim*	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
Freedom ERP	baixa	Sim	Não	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não
Oratio	média	Sim	Não	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
Sequoia ERP	baixa	Sim	Não	Sim	Sim	Não		Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
Tiny ERP	baixa	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não*	Sim
University ERP System	baixa	Não	Não	Sim	Sim	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
Value	média	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
WebERP	média	Sim	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim

* vide detalhes individuais do sistema

Quadro 34 - Características gerais dos aplicativos

6.7 Viabilidade de instalação nas pequenas empresas

De forma teórica, enquanto participantes dos projetos de *software* livre, as pequenas empresas têm as principais características descritas por Reis (2003, p.15) para suportar um projeto deste tipo:

- Possuem estrutura para acesso à internet;
- Tem tendência ao desenvolvimento próprio de aplicativos, agindo como desenvolvedor e usuário do próprio sistema que produziu;
- As pequenas empresas teriam interesse particular no sucesso do projeto e na implementação de melhorias;
- A presença do acesso à internet no ambiente da pequena empresa propicia condições de uso das ferramentas de comunicação (*e-mail*, *msn*) para utilizar e participar dos projetos de software livre.

Atendendo ao objetivo de estudo, esta seção compara as informações sobre os recursos de pessoas, equipamentos e aplicativos disponíveis nas pequenas empresas com as características apresentadas pelos aplicativos instalados para este trabalho.

6.7.1 Equipamento requerido

Uma das preocupações deste estudo foi procurar identificar se os recursos de *hardware* exigidos pelos aplicativos eram encontrados no ambiente da pequena empresa, pois o fato de exigências fora dos padrões que as pequenas empresas possuem poderia implicar na impossibilidade de instalação.

Os aplicativos de software livre obedecem os padrões de computadores do tipo PC. Todos os aplicativos instalados utilizam recursos de hardware que estão presentes no ambiente da pequena empresa, conforme demonstrado no Quadro 35, dispensando a necessidade de equipamentos específicos ou com capacidade e recursos cujos custos estariam fora do alcance das pequenas empresas.

Aplicativos	Hardware presente no ambiente da pequena empresa
Compiere	Sim
Evaristo	Sim
EzyBiz	Sim
Factura LUX	Sim
Freedom ERP	Sim
Oratio	Sim
Sequoia ERP	Sim
Tiny ERP	Sim
University ERP System	Sim
Value	Sim
WebERP	Sim

Quadro 35 – Hardware requerido e o ambiente da pequena empresa

Os recursos de rede de computadores, embora presentes nas pequenas empresas, não se mostraram como pré-requisito de instalação para nenhum dos aplicativos e todos os aplicativos funcionam mesmo que seja em uma única máquina.

6.7.2 Áreas funcionais

As informações apresentadas no capítulo quatro apontaram para o uso de aplicativos nas pequenas empresas, principalmente nas áreas comercial, produção, suprimentos e administrativa. Na seção 6.5.3, foram apresentados os módulos que foram encontrados nos sistemas instalados. Com base nestas duas informações é possível dizer que existe integração entre as áreas de utilização de sistemas integrados nas pequenas empresas e os recursos que são oferecidos pelos sistemas de software livre. O Quadro 36 mostra as áreas de utilização e os módulos que foram encontrados que atendem a cada uma das áreas.

Área de utilização	Módulos encontrados nos sistemas
Comercial	Vendas, clientes, CRM, pedidos
Produção	Manufatura, produtos
Suprimentos	estoque, fornecedores
Administrativa	Contabilidade, financeiro

Quadro 36 – Adequação módulos às áreas da pequena empresa

O quadro 36 apenas relaciona as áreas de utilização e os módulos encontrados, o que não quer dizer que é possível encontrar todos os módulos apresentados em um único sistema integrado. Como não existe um padrão, nem a obrigatoriedade da existência de módulos básicos, cada sistema pode combinar diferentes módulos. A escolha do sistema mais adequado à realidade da empresa depende da decisão da própria empresa, aliás é com base nestas características que o sistema é construído, normalmente para atender a um segmento ou atividade de forma mais específica.

Nenhum dos aplicativos instalados declarou estar pronto para atender à empresas de grande porte. O Compiere foi o único a declarar que o objetivo do aplicativo era atender à pequenas e médias empresas, ficando o restante vinculado à utilização em pequenas empresas.

6.7.3 Pessoas

Com a maioria dos sistemas tendo uma dificuldade de instalação classificada entre média e baixa, não parece existir barreiras que impeçam a instalação dos sistemas nas pequenas empresas. A instalação poderia ser feita com o próprio pessoal dedicado à área de TI. Também não seria difícil encontrar pessoas que tenham condições de realizar a instalação entre desenvolvedores de páginas *web* ou programadores, caso a empresa necessite contratar serviços de terceiros para a instalação.

A usabilidade verificada nos sistemas permite que a utilização seja feita por usuários com diferentes tipos de conhecimento e as características da interface entre usuários e sistema não foge aos padrões com os quais os usuários já tiveram contato em outros aplicativos. Todos os aplicativos fizeram uso do conceito de “janelas” com botões, caixas de texto e menus além do uso do *mouse* para seleção de opções. Assim, os usuários que precisem de algum treinamento ou adaptação em relação ao uso dos sistemas de software livre teriam a mesma necessidade em relação aos sistemas proprietários.

O idioma se apresenta como uma dificuldade. Apenas três dos aplicativos instalados tinham efetivamente a disponibilidade em português e a

menos que a pequena empresa que escolha os outros sistemas tenha em seu quadro de funcionários pessoas que conheçam o idioma no qual o sistema está disponível, ela vai encontrar dificuldades no treinamento e terá que propiciar um curso do idioma para seus funcionários ou ainda realizar a tradução do *software* para o português.

Sob o aspecto de alteração no código também é possível verificar que dentro de um objetivo de integração do sistema de informação com apoio da tecnologia aos objetivos estratégicos da empresa, dificilmente o sistema escolhido não precisará de alguma adequação. Encontrar pessoas para a realização destas alterações, mesmo que seja apenas do idioma conforme comentado no parágrafo anterior, implica em encontrar um profissional que compreenda a linguagem de programação, o ambiente sob o qual o *software* foi construído e aprender sobre ele, uma vez que nem sempre é possível encontrar suporte, mesmo que comercial (na maioria das vezes não disponível no Brasil para os aplicativos que não estão em português).

Assim, seria necessário encontrar ou formar pessoas capacitadas e inseridas no contexto do *software* livre que possam utilizar as formas de aprendizagem e suporte disponíveis neste modelo de desenvolvimento, fazendo uso de listas de discussão, *e-mail*, leitura de manuais e documentos, além da partilha de informações e conhecimento.

No caso do suporte disponível, a tendência é que seja mais fácil encontrar e formar profissionais que possam auxiliar a empresa na tarefa de adequação do *software*. Essa dificuldade com pessoas é reflexo de um dos pontos tidos como desvantagem do *software* livre comentada na seção 3.8.2.

O Quadro 37 apresenta os principais pontos favoráveis e desfavoráveis à instalação dos sistemas de *Software* Livre nas pequenas empresas.

Favorável	Desfavorável
<ul style="list-style-type: none"> ▪ Estrutura de <i>hardware</i> ▪ Relativa facilidade de instalação ▪ Acesso à internet ▪ Módulos dos sistemas estão em acordo com o uso nas pequenas empresas ▪ Foco de desenvolvimento dos sistemas de <i>software</i> livre está voltado para as pequenas empresas ▪ Cultura da pequena empresa em desenvolver seu próprio <i>software</i> 	<ul style="list-style-type: none"> ▪ Idioma (nem sempre disponível em português) ▪ Escassez de pessoas capacitadas à instalação em função das dificuldades relatadas com usuários nas pequenas empresas e a falta de pessoal capacitado em <i>software</i> livre ▪ Dificuldade de se encontrar suporte comercial no Brasil ▪ No caso de se encontrar suporte de terceiros (desenvolvedor de outro <i>software</i>) convencê-los a instalar o <i>software</i> livre ▪ Conceito de sistema integrado (ERP) ainda é pouco utilizado na pequena empresa.

Quadro 37 – Pontos favoráveis e desfavoráveis à instalação de *software* livre nas PE

Não foi possível dizer somente com a realização deste estudo se os resultados obtidos com a utilização do *Software* Livre como ferramenta estratégica em um sistema de informação, no caso da aplicação à pequena empresa levaria um tempo maior ou menor para adaptação do *software* à empresa em relação ao *software* proprietário.

Também não foi possível verificar se a utilização e a absorção de conhecimento da ferramenta, provocariam alguma espécie de ganho depois de sua implantação, nem tampouco se a filosofia do *software* livre traria mudanças para a formação das pessoas da área de TI ou mesmo à empresa como um todo. Essas questões são indicações para trabalhos futuros.

7 - Considerações finais

Este capítulo apresenta aspectos referentes ao texto de conclusão do trabalho e sugestões para trabalhos futuros.

7.1 Conclusão

Embora o termo ERP ainda represente uma categoria de sistemas mesmo em se tratando do *software* livre, pois era assim que se encontrava classificada a categoria no site *SourceForge.net* este trabalho optou por utilizar o termo sistema integrado de gestão por sua maior abrangência .

Existe um número crescente de projetos de sistemas integrados de gestão desenvolvidos sob os moldes do *software* livre, mas poucos deles (apenas 7%) está em estágio de produção estável. Estar neste estágio não significa que não tenha mais nada a ser acertado no aplicativo. Existem opções que ainda não foram implementadas, idiomas nos quais o aplicativo ainda não está disponível, mas está pronto para o uso em suas principais atribuições que já passaram pela fase de testes (beta).

Os aplicativos de *software* livre que foram instalados mostram-se bastante adaptáveis às pequenas empresas, com destaque para o fato de que nenhum deles declarou ter sido criado para atender as necessidades das grandes empresas.

A escassez de pessoal capacitado dificulta mas não impede que a instalação seja realizada, pois existe um ambiente propício para isso, uma vez que 56% das pequenas empresas possui área própria de TI.

Outra característica que apóia a instalação de *software* livre na pequena empresa é o fato de ser comum as pequenas empresas cuidarem do desenvolvimento de seu próprio *software*, seja por meio da contratação de serviços de terceiros, seja com seu próprio pessoal de TI.

A instalação e configuração foram classificadas como tendo um nível de dificuldade entre baixo e médio, embora em função da utilização das linguagens Java e PHP na maioria dos aplicativos não estão presentes utilitários de instalação pelo fato da necessidade de se instalar o interpretador da linguagem e o banco de dados. Em compensação o uso destas linguagens não impõe barreiras ao tipo de sistema operacional que a pequena empresa venha a utilizar ou mesmo que já utiliza em seu dia-a-dia.

O hardware necessário para instalação está presente no ambiente das pequenas empresas e não se mostrou necessário investimentos adicionais em características específicas de *hardware*, seja em função do modelo de computador necessário ou fabricante específico de equipamento.

Apenas um dos aplicativos não mostrou grande preocupação com a interface e o conceito de usabilidade, e os padrões de construção dos sistemas obedece à descrição dada por Sordi (2003) e apresentada no capítulo 2, quando o autor retrata uma interface baseada no conceito do ambiente gráfico e ambiente *web*.

Também ficou evidenciada a coerência entre a utilização de gerenciadores de bancos de dados desenvolvidos sob os moldes do *software* livre como elemento integrador de dados. Dos 13 aplicativos selecionados apenas um deles, o openXpertya, não utiliza um banco de dados *software* livre, embora haja projeto de fazê-lo.

Os aplicativos mostraram preocupação com questões de internacionalização, pois a maioria deles prevê sua utilização em mais de um idioma. Falta ainda prover uma maior flexibilidade quanto à adequação sobre as características de ordem fiscal e contábil que também variam de país para país. O idioma se mostrou uma restrição para o Brasil, pois apenas três dos sistemas tinham disponibilidade em português.

Os sistemas são compostos por diferentes módulos, o que representa a preocupação de cada sistema em atender a diferentes setores, embora alguns até nem se preocupem em declarar o vínculo de sua utilização à um determinado setor.

Não foi abordado neste trabalho a quantidade de pequenas empresas que utilizam *software* livre em suas atividades, mas as características dos sistemas estudados indicam adequação de seu uso muito mais nas pequenas

que nas grandes empresas, uma vez que estas provavelmente já passaram por um processo de escolha de um sistema integrado de gestão empresarial.

Uma dificuldade que pode ser encontrada é o fato de que na falta de pessoal capacitado para instalação, manutenção ou alteração do aplicativo, tendo a pequena empresa que recorrer a serviços de terceiros que provavelmente já são desenvolvedores de algum tipo de *software* e atuam no mercado com seu próprio produto, surja a dificuldade em convencer estes profissionais a trabalhar com o *software* livre.

Para o sucesso da utilização de um sistema de informação baseado no *software* livre, assim como no *software* proprietário, é necessária a existência de pessoal dedicado aos ajustes que se fizerem necessários para que o sistema possa realmente trazer vantagem competitiva à empresa. Assim fica indicada a necessidade da presença constante de pessoas comprometidas com este objetivo.

O fato da pequena empresa possuir pessoal dedicado à área de informática, ainda que de forma não estruturada ou acumulando funções, e ter como característica o desenvolvimento próprio de seus aplicativos pode ajudar a superar esta dificuldade e auxiliar a absorção de tecnologia, mas é preciso lembrar novamente que isto não dispensa a necessidade da presença de uma equipe envolvida com a proposta de desenvolvimento e adequação do *software* à empresa.

Informações obtidas no capítulo 4 apontam para uma baixa utilização dos sistemas integrados de gestão nas pequenas empresas que parecem ainda estar voltadas para o uso de sistemas isolados em seus departamentos. Este fato pode funcionar tanto como um ponto negativo se consideradas as dificuldades em realizar esta integração, como um ponto positivo, se considerado que existe muita necessidade a ser suprida à medida que as pequenas empresas adotem o conceito de integração, motivadas pelo uso destes sistemas nas grandes empresas, e pelos fabricantes que estão se voltando para a atuação no mercado das médias e pequenas empresas, seja com aplicações específicas, seja com a utilização do conceito ASP (*Application Service Provider*).

Este trabalho não considerou outras limitações para o uso da tecnologia nas pequenas empresas, tais como os citados por Moraes (2005): resistência

humana à implantação, falta de adequação precisa das informações, perdas financeiras relacionadas a projetos de tecnologia que não deram certo, retorno não satisfatório sobre investimento ou altos preços. Apesar disso, é possível compreender que a pequena empresa que optar pelo uso de um sistema de gestão que tenha sido desenvolvido sob os moldes do *software* livre deve aceitar o desafio quanto à adequação do *software*, investimento em formação e capacitação de profissionais desenvolvedores e usuários. Embora a maioria destes requisitos seja descrita pela literatura como necessária também para a utilização de *software* proprietário, no caso do *software* livre a absorção da tecnologia é fator intrínseco ao desenvolvimento e adequação do *software* por membros da própria empresa face à dificuldade de se conseguir pessoal capacitado, tanto no que se refere às pequenas empresas quanto ao *software* livre.

Como a amostra utilizada para esta pesquisa não foi probabilística, não pode ser utilizada como elemento representativo para o universo de todos os aplicativos de *software* livre, mas os resultados aqui apresentados podem ajudar a explicar e descrever diversos aspectos que contribuem para a compreensão do fenômeno do *software* livre, bem como servir de base para indicadores utilizados em trabalhos futuros.

7.2 Sugestões para estudos futuros

Em se tratando de dois temas considerados ainda pouco explorados por pesquisas, tanto o *Software* Livre quanto Pequenas Empresas dão perspectivas para diversas possibilidades de estudo.

A seguir, são apresentadas algumas destas perspectivas que surgiram de questionamentos e possibilidades identificadas durante este trabalho, embora ainda não estejam concretizadas sob os aspectos metodológicos:

- **Verificação do comportamento dos sistemas após a instalação:** este trabalho se concentrou na instalação dos aplicativos, mas não simulou ou vivenciou um ambiente de produção com o *software* em pleno funcionamento. Trabalhos deste tipo poderiam verificar a estabilidade, adaptação de usuários e outros aspectos.

- **Comparativo de funcionalidades entre um aplicativo proprietário e *software* livre:** considerando a utilização do *software* livre, comparar suas funcionalidades à um software proprietário que lhe seja equivalente. Por exemplo, Appelbe (2003) cita como concorrentes o aplicativo Compiere e o proprietário Peoplesoft.
- **Comparativo entre funcionalidades de módulos de aplicativos:** seguindo o mesmo raciocínio do item anterior, outra indicação de trabalho seria a verificação de funcionalidades entre módulos de sistemas diferentes, identificando as semelhanças e diferenças, objetivando um estudo mais detalhado que permitiria maiores informações para que o usuário possa compara-las com sua necessidade de uso, principalmente sob o ponto de vista da gestão da produção.
- **Comparativo de custos e prazo de adequação de sistemas de *software* livre e proprietário:** este trabalho, de forma mais ampla não se concentraria no comparativo de dois aplicativos, mas teria como objetivo a identificação das diferenças, das dificuldades e facilidades das adequações de sistema de *software* livre em comparação com sistemas proprietário. As metodologias e técnicas de implantação de sistemas integrados estão de certa forma bastante descritas na literatura, mas será que o software livre obedece ao que já foi descrito?
- **Estudo de usabilidade de software livre sobre a ótica de outros métodos de avaliação:** o método de usabilidade utilizado para este trabalho (Heurística) não é o único existente. Os trabalhos de Nichols e Twidale (2005) e Reis (2003) apontam para uma baixa usabilidade, apesar de não estarem voltados para o tipo de aplicativo analisado neste trabalho, esta característica não foi confirmada neste trabalho em particular. Assim caberia um estudo mais detalhado e com foco apenas na usabilidade, talvez até com mais de um método para a verificação da realidade atual da usabilidade do *software* livre de forma geral e do software livre da categoria “sistema integrado de gestão”.
- **Aplicar testes aos sistemas integrados de gestão de software livre:** a ciência da computação dispões de métodos e técnicas específicas

para se aplicar testes em *software* (teste de stress, caixa preta). Estes métodos poderiam ser aplicados para identificar o comportamento dos software livre.

- **Estudo sobre a tecnologia ASP na pequena empresa:** a utilização do ASP em pequenas empresas poderia ser estudada sobre aspectos de casos de uso, adequação, levantamento de problemas e custos, bem como do impacto do uso desta tecnologia sobre a estratégia da empresa.
- **Suporte técnico – aspectos de aprendizagem:** o suporte à utilização, manutenção e instalação no *software* livre é realizado por meio de grupos de discussão, documentação e troca de informações pessoais, por *e-mail* ou outras ferramentas de comunicação. Será que este tipo de “suporte” determina um ganho de aprendizagem e capacitação? Quais são suas falhas? O tempo necessário para absorver o aprendido é compensatório em relação à um suporte comercial?
- **Desenvolvimento de sistema integrados de gestão a partir de modelos (framework) :** durante a fase de seleção de amostras foram encontrados programas classificados como “frameworks” que não são sistemas prontos, mas são utilitários que fornecem modelos a partir dos quais é possível se criar módulos e conseqüentemente sistemas inteiros que podem ser utilizados pelas empresas. Sob este aspecto surgem questões sobre a melhor adequação desta opção à seleção de um sistema que já esteja “pronto”.
- **O uso da filosofia do software livre traz mudanças ao comportamento da pequena empresa como um todo? :** utilizar software livre parece implicar em uma necessidade de absorver a tecnologia para depois poder adaptá-la às necessidades. Será que esta filosofia é algo que acaba “contaminando” outros aspectos da pequena empresa ou fica restrita à área de TI?

BIBLIOGRAFIA

APPELBE, B. (2003). The future of open source software. **Journal of Research and Practice in Information Technology**, Sydney, v.35, n.4, p.227-235, Nov.

ASSOCIAÇÃO EMPRESARIAL DE PORTUGAL (2005). Disponível em:<<http://www.aeportugal.pt/Inicio.asp?Pagina=/Aplicacoes/Noticias/Noticia&Codigo=5250>>. Acesso em: 14 jan. 2006.

AUGUSTO, M.P. (2003). **Um estudo sobre as motivações e orientações de usuários e programadores brasileiros de software livre**. Dissertação (Mestrado) - Instituto COPPEAD de Administração, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003

BARBIERE, J.C. (1999). **Produção e transferência de tecnologia**. São Paulo: Ática

BARCELOS, I.F. (2001). **Aplicação de um processo de usabilidade em uma ferramenta de suporte à escrita técnica em inglês**. 126p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2001.

BARR, J. (2001). Live and let licence. A primer on software licensing in the Open Source context. **LinuxWorld**. Disponível em: <<http://www.itworld.com/AppDev/350/LWD010523vcontrol4/pfindex.html>>. Acesso em: 08 fev 2006.

BASTOS, A. et al. (2005). **Cartilha de software livre**. 2.ed. Bahia: Projeto de Software Livre. Disponível

em:<<http://twiki.im.ufba.br/pub/PSL/ImpressaoDaCartilha/cartilha.pdf>>. Acesso em: 20 fev. 2006.

BEHLENDORF, B. (1999). Open source as a business strategy. In: OPEN sources (computer file): voices from the open source revolution. Beijing: O'Reilly. p.149-170.

BERALDI, L.C.; ESCRIVÃO FILHO, E.; RODRIGUES, D.M. (2000). Avaliação da adequação do uso de tecnologia de informação na pequena empresa. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 6., 2000, Bauru. **Anais eletrônicos...** Bauru: UNESP. Disponível em:<<http://simpep.feb.unesp.br>>. Acesso em: 10 jan. 2005.

BERNARDO, J.J. (1999). **Sistemas de informação na gestão empresarial**: estudo dos sistema de informação que suportam a tomada de decisão em diversos níveis gerenciais, das pequenas empresas industriais da região de São José do Rio Preto - SP. 103p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1999.

BEST SOFTWARE (2004). **A complete software evaluation guide for small and medium-size wholesaler-distributor**. Technology Evaluation Centers. Disponível em:<<http://simpep.feb.unesp.br>>. Acesso em: 5 Jan. 2006.

BIGATON, A.L.W. (2005). **Gestão estratégica da informação nas pequenas empresas**: estudo comparativo de casos em empresas do setor industrial de São José do Rio Preto – SP. 200p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2005.

BIO, S.R. (1996). **Sistemas de informação**: um enfoque gerencial. São Paulo:Atlas.

BLATTMANN, U.; TRISÃO, A.M. (2005). Internet como instrumento de pesquisa técnico-científica na engenharia civil. **Revista ACB**, v.4, n.4, p.28-46, ago. Disponível em:<<http://www.acbsc.org.br/revista/ojs/viewarticle.php?id=33>>. Acesso em: 27 fev. 2006.

BRASIL (2004). **Constituição da República Federativa do Brasil**: texto constitucional promulgado em 5 de outubro de 1988, com as alterações adotadas pelas ementas constitucionais nº 1/92 a 42/2003 e pelas ementas constitucionais de revisão nº 1 a 6/94. Brasília, DF: Senado Federal.

_____. (2005). Lei 11.196, de 21 de novembro de 2005. Institui o regime especial de tributação para a plataforma de exportação de serviços de tecnologia da informação - REPES, o regime especial de aquisição de bens de capital para empresas exportadoras - RECAP e o programa de inclusão digital; dispõe sobre incentivos fiscais para a inovação tecnológica; altera o decreto-Lei nº 288, de 28 de fevereiro de 1967, o decreto nº 70.235, de 6 de março de 1972, o decreto-lei nº 2.287, de 23 de julho de 1986, as leis nºs 4.502, de 30 de novembro de 1964, 8.212, de 24 de julho de 1991, 8.245, de 18 de outubro de 1991, 8.387, de 30 de dezembro de 1991, 8.666, de 21 de junho de 1993, 8.981, de 20 de janeiro de 1995, 8.987, de 13 de fevereiro de 1995, 8.989, de 24 de fevereiro de 1995, 9.249, de 26 de dezembro de 1995, 9.250, de 26 de dezembro de 1995, 9.311, de 24 de outubro de 1996, 9.317, de 5 de dezembro de 1996, 9.430, de 27 de dezembro de 1996, 9.718, de 27 de novembro de 1998, 10.336, de 19 de dezembro de 2001, 10.438, de 26 de abril de 2002, 10.485, de 3 de julho de 2002, 10.637, de 30 de dezembro de 2002, 10.755, de 3 de novembro de 2003, 10.833, de 29 de dezembro de 2003, 10.865, de 30 de abril de 2004, 10.925, de 23 de julho de 2004, 10.931, de 2 de agosto de 2004, 11.033, de 21 de dezembro de 2004, 11.051, de 29 de dezembro de 2004, 11.053, de 29 de dezembro de 2004, 11.101, de 9 de fevereiro de 2005, 11.128, de 28 de junho de 2005, e a medida provisória nº 2.199-14, de 24 de agosto de 2001; revoga a lei nº 8.661, de 2 de junho de 1993, e dispositivos das leis nºs 8.668, de 25 de junho de 1993, 8.981, de 20 de janeiro de 1995, 10.637, de 30 de dezembro de 2002, 10.755, de 3 de novembro de 2003,

10.865, de 30 de abril de 2004, 10.931, de 2 de agosto de 2004, e da medida provisória nº 2.158-35, de 24 de agosto de 2001; e dá outras providências. Disponível em: <https://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2005/Lei/L11196.htm>. Acesso em: 14 jan. 2006.

BORGES, C. GEYER, C.F.R. (2003). Estratégias de governo para promover o desenvolvimento de *software* livre. **Revista RESI**, a.2, v.2, n.1, jul. Disponível em: <<http://server.presidentekennedy.br/resi/edicao02/artigo05.pdf>>. Acesso em: 08 fev 2006.

CAMPOS, R.R.; CAZARINI, E.W. (2005a). Sistemas ERP de código aberto: uma opção para as pequenas indústrias?. In: CONGRESSO INTERNACIONAL DE GESTÃO DA TECNOLOGIA E SISTEMAS DE INFORMAÇÃO, 2., 2005, São Paulo. **Anais...** São Paulo: TECSI/EAC/FEA/USP. 1 CD-ROM.

_____. (2005b). Integrando o sistema de informação gerencial à organização: aspectos da modelagem organizacional segundo a metodologia EKD. In: CONGRESSO BRASILEIRO DE SISTEMAS, 2005, Ribeirão Preto. **Anais...** Ribeirão Preto: USP. 1 CD-ROM.

CAUTELA, A.L.; POLLONI, E.G.F. (1996a). **Sistemas de informação na administração de empresas**. 4.ed. São Paulo: Atlas.

_____. (1996b). **Sistemas de informação: técnicas avançadas de computação**. 4.ed. São Paulo: Atlas.

CERRI, M.L. (2004). **Enterprise Resource Planning**: um estudo sobre estratégias de implantação. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.

COLANGELHO FILHO, L. (2001). **Implantação de sistemas ERP (Enterprise Resources Planning)**: um enfoque de longo prazo. São Paulo: Atlas.

COLES, S.; ROWLEY, J. (1996). **Spreadsheet modeling for management decision making** – industrial management & data system. Dunfermline: MCB University Press. p.17-23.

COMPIERE ERP & CRM (2003). **Quienes somos**. Disponível em:<<http://www.compiere.com.ar/home/aboutus.html>>. Acesso em: 28 feb. 2006.

CORRÊA, H.C. (1998). ERP's: por que as implantações são tão caras e raramente dão certo?. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INDUSTRIAIS, 1998, São Paulo. **Anais...** São Paulo: FGV. p.288-300

CORRÊA, H.L.; GIANESI, I.G.N. (1996). **Just in time, MRP II e OPT** – um enfoque estratégico. 2.ed. São Paulo: Atlas.

CORRÊA, H.L.; GIANESI, I.G.N.; CAON, M. (2001). **Planejamento, programação e controle da produção: MRP II/ERP - conceitos, uso e implantação**. 4.ed. São Paulo: Atlas.

CRUZ, T. (2003). **Sistemas de informações gerenciais: tecnologias da informação e a empresa do século XXI**. 3.ed. São Paulo: Atlas.

DANE, F.C. (1990). **Research methods**. Belmont: Brooks/Cole.

DAVENPORT, T.H. (2002). **Missão crítica: obtendo vantagem competitiva com os sistemas de gestão empresarial**. Tradução de Raul Rubenich. Porto Alegre:Bookman.

FEDERAÇÃO DAS INDÚSTRIAS DO ESTADO DE SÃO PAULO; FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE DA USP (2004). **Idigital** - perfil da empresa digital 2003/2004. Disponível em:<<http://www.idigital.fea.usp.br/iDigital/Repositorio/0/Documentos/iDigital2004.pdf>>. Acesso em: 6 jan. 2006.

FREE SOFTWARE FOUNDATION (2004). **O que é software livre**. Disponível em:<<http://www.gnu.org/philosophy/free-sw.pt.html>>. Acesso em: 28 dez. 2004.

FREE SOFTWARE FOUNDATION EUROPE (2004). **Porque falamos de software livre**. Disponível em:<<http://fsfeurope.org/documents/whyfs.pt.html>>. Acesso em: 29 dez. 2004.

FREITAS, H. et al. (2004). **Perfil da tecnologia da informação no Rio Grande do Sul**: grandes empresas, pequenas empresas e cooperativas. Porto Alegre: PPGA/EA/UFRGS. Disponível em:<http://professores.ea.ufrgs.br/hfreitas/revista/arquivos/edicao_internet/ago_to_04/gianti_hf_cdpm_h15.pdf>. Acesso em: 18 jan. 2006.

FRESHMEAT (2005). **About freshmeat**. Disponível em:<<http://freshmeat.net/about/>>. Acesso em: 3 Jan. 2005.

GIL, A.C. (1998). **Métodos e técnicas de pesquisa social**. São Paulo: Atlas.

GIL, A.L. (1999). **Sistemas de informações contábil/financeiros**. 3.ed. São Paulo: Atlas.

HABERKORN, E. M. (1999). **Teoria do ERP – Enterprise Resource Planning**. São Paulo: Makron Books.

_____. (2003). **Gestão empresarial com ERP**. São Paulo: Microsiga Software.

HEHN, H.F. (1999). **Peopleware**: como trabalhar o fator humano nas implementações de sistemas integrados de informação (ERP). São Paulo: Gente.

HENDERSON, B.D. (1989). The origin of strategy. In: STRATEGY: seeking and securing competitive advantage. Boston: Harvard Business School. p.3-9. (The Harvard business review book series).

HEXSEL, R.A. (2002). **Software livre**: propostas de ações de governo para incentivar o uso de software livre. Curitiba: Universidade Federal do Paraná. (Relatório técnico n. RT-DINF 004/2002). Disponível em: <http://www.inf.ufpr.br/info/techrep/RT_DINF004_2002.pdf>. Acesso em 05 fev. 2006

HOHMAN, R. (1999). **GNU's not unix**. Tradução de Erik Kohler. Disponível em:<http://www.geocities.com/CollegePark/Union/3590/gnus_not_unix.html>. Acesso em: 3 jan. 2005.

JAKOVLJEVIC, P.J. (2000). Essencial ERP – its underpinning technology. Technology Evaluation Centers. Disponível em:<http://www.technologyevaluation.com/login.asp?l=/tec.asp&name=article&article=TU_BA_PJ_09_05_05_1.asp&category=BusinessApplications&path=/Research/ResearchHighlights/BusinessApplications/2005/09/research_notes/TU_BA_PJ_09_05_05_1.asp?id=64.2005.09.05.3170&e=mr.fields%2Citelefonica.com.br?id=64.2005.09.05.3170&e=mr.fields%2Citelefonica.com.br>. Acesso em: 5 Jan. 2006.

JOBIM FILHO, P. (1979). **Uma metodologia para o planejamento e o desenvolvimento de sistemas de informação**. São Paulo: Edgard Blücher.

KALPIC, D.; FERTALJ, K. (2004). ERP software evaluation and comparative analysis. **Journal of Computing and Information Technology**, Zagreb, v.12, n.3, p.195-209, Sept.

KRISHNAMURTHY, S. (2002). Cave or community. **First Monday: peer-reviewed journal on the internet**. v.7, n.6, jun. Disponível em: <http://www.firstmonday.dk/issues/issue7_6/krishnamurthy/>. Acesso em 09 fev. 2006.

KON, F. (2001). O *software* aberto e a questão social. São Paulo. Universidade de São Paulo. (Relatório Técnico n. RT-MAC-2001-07). Disponível em: <<http://www.ime.usp.br/~kon/papers/RT-SoftwareAberto.pdf>>. Acesso em: 11 fev. 2006

LAUDON, K.C.; LAUDON, J.P. (2001). **Gerenciamento de sistemas de informação**. 3.ed. Rio de Janeiro: LTC.

LEME FILHO, T. (2003). **Metodologia de desenvolvimento de sistemas**. Rio de Janeiro. Axel Books do Brasil Editora.

LENNER, J.; TIROLE, J. (2000). *The simple economics of Open Source*. **NBER Working Paper Series** (National Bureau of Economic Research). Disponível em: <<http://www.nber.org/papers/w7600>>. Acesso em: 08 fev. 2006

LICZBINSKI, C.R. (2002). **Modelo de informações para o gerenciamento das atividades das pequenas indústrias de produtos alimentares do Rio Grande do Sul**. 158p. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Florianópolis, 2002.

LONGENECKER, J.G.; MOORE, C.W.; PETTY, J.W. (1997). **Administração de pequenas empresas: ênfase na gerência empresarial**. Tradução de Maria Lucia G. L. Rosa e Sidney Stancatti. São Paulo: Makron Books.

LOSSO, M.E.F. (2004). Noções de direito autoral e sua regulamentação internacional. **Jus Navigandi**, Teresina, ano 8, n.464, out. Disponível em:<<http://jus2.uol.com.br/doutrina/texto.asp?id=5806>>. Acesso em: 2 fev. 2006.

MAÑAS, A.V. (1999). **Administração de sistemas de informação**. São Paulo: Érica.

MARCONI, M.A.; LAKATOS, E.M. (1999). **Técnicas de pesquisa: planejamento e execução de pesquisas, amostragens e técnicas de pesquisa,**

elaboração, análise e interpretação de dados. 4.ed. São Paulo: Atlas.

MARTINS, G.A. (1994). **Epistemologia da pesquisa em administração**. Tese (Livre docência) - Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, São Paulo, 1995.

MCFARLAN, F.W. (1979). Information technology changes the way you compete. In: STRATEGY: seeking and securing competitive advantage. Boston: Harvard Business School. p.77-88. (The Harvard business review book series).

MCGEE, J.; PRUSAK, L. (1994). **Gerenciamento estratégico da informação**. 13.ed. Rio de Janeiro: Campus.

MEIRELES, F.S. (2005). Administração de recursos de informática. In: 16ª PESQUISA anual, 2005 - administração de recursos de informática. Disponível em:<<http://www.eaesp.fgvsp.br/subportais/interna/Sobre/Pesq04GV.pdf>>. Acesso em: 7 jan. 2006.

MENDES. J.V.; ESCRIVÃO FILHO, E. (2002). Sistemas integrados de gestão ERP em pequenas empresas: um confronto entre o referencial teórico e a prática empresarial. **Revista Gestão & Produção**, São Carlos, v.9, n.3, p.277-296, dez.

MENDES, J.V. (2003). **Avaliação de sistemas ERPs como ferramenta de mudança organizacional nas pequenas e médias empresas** – um roteiro auxiliar. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2003

MORAES, G.D.A. (2005). **A tecnologia da Informação na pequena empresa**: uma investigação sobre sua contribuição à gestão estratégica da Informação nos empreendimentos industriais de São José do Rio Preto – SP. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2005.

- NICHOLS, D.M.; TWIDALE, M.B. (2005). **The usability of open source software**. Disponível em:
<http://firstmonday.org/issues/issue8_1/nichols/index.html>. Acesso em: 5 Jan. 2005.
- NIELSEN, J. (1993). **Usability engineering**. Londres: Academic Press.
- OLIVEIRA, D.P.R. (2004). **Sistemas de informações gerenciais: estratégicas, táticas, operacionais**. 9.ed. São Paulo: Atlas.
- OPEN SOURCE INITIATIVE (OSI) (2004). **The open source definition – version 1.9**. Disponível em:<<http://www.opensource.org/docs/definition.php>>. Acesso em: 29 Dec. 2004.
- PADILHA, T.C.C.; MARINS, F.A.S. (2005). Sistemas ERP: características, custos e tendências. **Revista Produção**, São Paulo, v.15, n.1, p.102-113, jan./abr.
- PALERMO, F.K.O. (2003). Elementos de comparação entre copyright e direito do autor. **Jus Navigandi**, Teresina, ano 7, n.62, fev. Disponível em:<<http://jus2.uol.com.br/doutrina/texto.asp?id=3757>>. Acesso em: 2 fev. 2006.
- PEREIRA, I. (2004). O movimento do *software* livre. In: CONGRESSO LUSO-AFRO-BRASILEIRO DE CIÊNCIAS SOCIAIS, 8, 2004, Coimbra – Portugal. **Anais...** Coimbra: Centro de estudos Sociais, Faculdade de Economia, Universidade de Coimbra. Disponível em:
<<http://www.ces.uc.pt/lab2004/pdfs/lnesPereira.pdf>>. Acesso em: 10 fev. 2006.
- PERENS, B. (1999). The open source definition. In: OPEN sources (computer files): voices from the open source revolution. Beijing: O'Reilly. p.171-188.
- PORTER, M.E. (1979). How competitive forces shape strategy. In: STRATEGY: seeking and securiting competitive advantage. Boston: Harvard

Business School. p.11-26. (The harvard business review book series).

_____. (1980). **Competitive strategy**. New York: Free Press.

REIS, C.R. (2003). **Caracterização de um processo de software para projetos de software livre**. 158p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2003.

REIS, A.P.; ESCRIVÃO FILHO, E. (2005). Análise das barreiras existentes para a prática do treinamento na pequena empresa. In: PREVIDELLI, J.J.; MEURER, V. (Org.). *Gestão da micro, pequena e média empresa no Brasil: uma abordagem multidimensional*. Maringá: Unicorpore.

RAIMOND, E. S. (1998). *The cathedral and the bazaar*. Tradução de Erik Kholer. Disponível em: <<http://www.geocities.com/CollegePark/Union/3590/pt-cathedral-bazaar.html>>. Acesso em: 11 fev. 2006.

REZENDE, D.A.; ABREU, A.F. (2000). **Tecnologia da informação aplicada a sistemas de informação empresariais**. São Paulo: Atlas.

RITZMAN, L.; KRAJEWSKI, L.; MOURA, R. (1989). **MRP, MRPII, MRP III (MRP + JIT com Kanban)**. São Paulo: IMAM.

RODRIGUES, W.J. et al. (2001). Free software in superior education: talking about *freedom* to future professionals. In: WORKSHOP SOBRE SOFTWARE LIVRE, 2., 2001, Porto Alegre. **Anais...** PortoAlegre: Sociedade Brasileira de Computação. p.43-45.

SALEH, A.M. (2004). **Adoção de tecnologia**: um estudo sobre o uso de software livre nas empresas. 148p. Dissertação (Mestrado) – Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, São Paulo, 2004.

SANDER, P.V. (2004). **Definição de open source** – version 1.0. Disponível em:<<http://people.deas.harvard.edu/~pvs/osd/po-osd.html>>. Acesso em: 28 Dec. 2004.

SANDRINI, J. (2006). Deu na folha: Lojas acirram disputa por computador PC popular. Projeto *Software Livre* Brasil. Disponível em: <<http://www.softwarelivre.org/news/5644>>. Acesso em: 11 fev. 2006.

SCHROEDER, U.; RODRIGUES, L.C. (2002). Sistema de Informação como recurso estratégico em empresas da região de Blumenau. **Revista de Negócios**, Blumenau, v.7, n.4, p.7-20, out./dez.

SERVIÇO BRASILEIRO DE APOIO ÀS MICRO E PEQUENAS EMPRESAS (2003). **A informatização nas MPES paulistas**. Disponível em:<www.sebraesp.com.br>. Acesso em: 1 set. 2003.

_____. (2004). **As MPES na economia**. Disponível em:<www.sebraesp.com.br>. Acesso em: 6 jan. 2005.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. (2002). **Administração da produção**. 2.ed. São Paulo: Atlas.

SCHWINGEL, C. (2003). Ciberativismo: o movimento *software* livre. In: Congresso brasileiro de ciência da comunicação, 26., 2003, Belo Horizonte. **Anais...** Belo Horizonte: Sociedade Brasileira de Estudos Interdisciplinares da comunicação. Disponível em: <<http://reposcom.portcom.intercom.org.br/dspace/bitstream/1904/4756/1/NP8SCHWINGEL.pdf>>. Acesso em: 07 fev. 2006.

SMALL BUSINESS ADMINISTRATION (2005). **United States small business administration**. Disponível em:<<http://www.sba.gov/size/summary-what-is.html>>. Acesso em: 14 Jan. 2006.

SOFTWARELIVRE (2005). **SoftwareLivre**. Disponível em:<<http://www.softwarelivre.gov.br>>. Acesso em: 11 jan. 2006.

SORDI, J. O.(2003). **Tecnologia da Informação aplicada aos negócios**. São Paulo: Atlas.

SOURCE FORGE (2005). **What is sourceforge.net?**. Disponível em:<<http://sourceforge.net/docman/>>. Acesso em: 3 Jan. 2005.

SPREMIC, M.; VUKSIC, V.B. (2005). ERP system Implementation and business process change: case study of a pharmaceutical company. **Journal of Computing and Information Technology**, Zagreb, v.13, n.1, p.11-24, Mar.

STALLMAN, R.M. (1991). **15 years of free software** – Quinze anos de software livre. Tradução de Erick Kohler. Disponível em:<<http://www.geocities.com/CollegePark/Union/3590/15anossoftwarelivre.html>>. Acesso em: 3 Jan. 2005.

_____. (1996). **Linux and the GNU system**. Tradução de Erik Kohler. Disponível em:<<http://www.geocities.com/CollegePark/Union/3590/linuxgnu.html>>. Acesso em: 3 Jan. 2005.

_____. (1998). **Linux e o projeto GNU**. Disponível em:<<http://www.gnu.org/gnu/linux-and-gnu.pt.html>>. Acesso em: 30 dez. 2004.

_____. (1999). The GNU operating system and the free software movement. In: OPEN sources (computer files): voices from the open source revolution. Beijing: O'Reilly. p.53-70.

STAIR, R.M.; REYNOLDS, G.W. (1999). **Princípios de sistemas de informação**. 4.ed. Rio de Janeiro: LTC.

TREVISAN, N. (2005). **Princípio da inclusão digital**: por mares nunca dantes

navegados - estudos para a inclusão de baixa renda na sociedade da informação. 126p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2005.

TORRES, N.A. (1995). **Competitividade empresarial com a tecnologia de informação**. São Paulo: Makron Books.

WARD, J.; PEPPARD, J. (2002). **Strategic planning for information systems**. 3.ed. New York: John Wiley.

WIKIPEDIA: the free encyclopedia. (2005). **LAMP (software bundle)**.

Disponível em:<http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29>.

Acesso em: 16 Feb. 2006.

VIDAL, A.G.R. (2005). **Aquisição de software e serviços de informática na pequena e média empresa**. Disponível

em:<<http://www.idigital.fea.usp.br/iDigital/Repositorio/0/Documentos/aquimpe.pdf>>. Acesso em: 3 jan. 2006.

YIN, R.K. (2001). **Estudo de caso: planejamento e métodos**. Porto Alegre:Atlas. Cap.1, p.20-37.

ZIMMERER, T.W.; SCARBOROUGH, N.M. (1994). **Essential of small business management**. New York: Macmillan College. Cap.16, p.441–458.

Caso não possua instalador, descrever forma de instalação (compilar fontes, descompactar arquivos...)

Nível de dificuldade de instalação: () Baixo () Médio () Alto

SOFTWARE ADICIONAL

Existe necessidade da instalação de *software* adicional?

() SIM () Não

Quais?

Descrever o processo de instalação dos *softwares* adicionais, verificando dificuldades de instalação e nível de conhecimento necessário para sua realização.

Nível de dificuldade de instalação: () Baixo () Médio () Alto

2 - FUNCIONALIDADES

INFORMAÇÕES SOBRE BANCO DE DADOS

SGBD nativo:
Possibilita trabalhar com outro SGBD? <input type="checkbox"/> SIM <input type="checkbox"/> Não
Quais?
Forma de acesso ao SGBD nativo (ODBC, biblioteca):
Forma de acesso a outros SGBD:
É possível importar dados existentes em outras bases? <input type="checkbox"/> SIM <input type="checkbox"/> Não
Quais as formas possíveis de importação (arquivo texto, outros formatos):
É possível criar ou alterar campos e tabelas existentes para adaptar a base às necessidades da empresa? <input type="checkbox"/> SIM <input type="checkbox"/> Não
As alterações feitas na base de dados podem ser ajustadas nas telas do sistema? <input type="checkbox"/> SIM <input type="checkbox"/> Não
É possível gerenciar dados de uma de uma empresa (ex: matriz e filial) ? <input type="checkbox"/> SIM <input type="checkbox"/> Não

RELATÓRIOS E GRÁFICOS

Quais são os relatórios oferecidos pelo <i>software</i> , de acordo com a instalação padrão?
Quais são os formatos sob os quais os relatórios são apresentados? <input type="checkbox"/> na tela do computador <input type="checkbox"/> apenas na impressora <input type="checkbox"/> na tela e na impressora
É possível gravar os relatórios em arquivos? () SIM () Não
Quais são os formatos de arquivo que os relatórios podem ser gravados?
Existem ferramentas que permitam criar novos relatórios ou alterar os relatórios existentes para adaptá-los à necessidade da empresa? () SIM () Não
É possível gerar gráficos com as informações? <input type="checkbox"/> SIM () Não
Quais tipos de gráfico podem ser gerados?

RECURSOS DE INTEGRAÇÃO COM OUTROS *SOFTWARES*

<i>Software</i> possibilita integração com outros <i>softwares</i> ? <input type="checkbox"/> SIM () Não
Quais?

Nível de dificuldade de integração: () Baixo () Médio () Alto
Módulos disponíveis no sistema:

RECURSOS DE COMUNICAÇÃO PRESENTES NO SOFTWARE

Software possui ferramentas de comunicação próprias (<i>e-mail</i> , messenger)? () SIM () Não
Quais?

SEGURANÇA

Existe senha para acesso ao <i>software</i> ? () SIM () Não
Existem níveis de acesso (permissão) diferentes ? () SIM () Não
Descrição dos níveis de permissão:

Que tipo de segurança podem ser impostas pelo <i>software</i> ? (criptografia, segurança do gerenciador de banco de dados, segurança do próprio sistema operacional)
Qual é a política de <i>backup</i> oferecida pelo <i>software</i> ?

3 - VIABILIDADE DE USO

Interface é simples e natural, promovendo interação com usuário (tela simples e visual agradável de forma a facilitar compreensão)? () SIM () Não
Existem suporte a diferentes tipos de idioma? () SIM () Não Quais?
Linguagem utilizada em menus e diálogos é expressa de forma clara, utilizando expressões habituais do usuário? () SIM () Não

Botões, objetos, ações e opções são apresentadas de forma fácil e visível? () SIM () Não
Sistema fornece informações constantes ao usuário sobre o que está sendo feito, e de que forma a informação fornecida pelo usuário foi interpretada? () SIM () Não
Sistema fornece indicações claras dos resultados, facilitando o controle sobre o sistema (ex: opções de desfazer)? () SIM () Não
Existe presença de "atalhos" que possam acelerar interação para os usuários mais experientes? () SIM () Não
As mensagens de erro são apresentadas com clareza, evitando utilização de códigos obscuros de erros? () SIM () Não
Existe um mecanismo de detecção de erros e procura de minimização de erros ? () SIM () Não
Existe um mecanismo de ajuda e solução de dúvidas quanto á utilização do <i>software</i> ? () SIM () Não
Existe documentação disponível sobre o <i>software</i> ? () SIM () Não
A documentação detalha instalação e utilização? () SIM () Não
Qual a forma de acesso à documentação (diretamente do sistema, via <i>web</i>)?
Além da documentação, existe alguma forma de suporte à instalação e utilização ou treinamento? () SIM () Não Explicar:

APÊNDICE B – RELAÇÃO DE *SOFTWARES*

Sistemas Integrados de Gestão – *SourceForge*

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
1	ABS							
2	Alamak ERP	NC	NC	NC	NC	NC	NC	NC
3	Alternative ERP	NC	NC	PHP	NC	NC	NC	NC
4	Athena Business Development	NC	NC	NC	NC	NC	NC	NC
5	BasicGes	IN	PN	Visual Basic	Windows	Access MySQL	PC	Espanhol
6	Benterprise	NC	NC	NC	NC	NC	NC	NC
7	Blue Project	NC	NC	NC	NC	NC	NC	NC
8	bubica	NC	NC	NC	NC	NC	NC	NC
9	Bussines Maker ERP	IN	NC	NC	NC	NC	NC	NC
10	BUSINESS PLAN	NC	NC	NC	NC	NC	NC	NC
11	Compiere	IN	PS	Java	Linux Windows MAC		PC / MAC	Inglês, Catalão, Alemão, Espanhol, Francês, Finlandês, Italiano, Japonês, Neozelandês, Polonês, Português (Brasil), Russo, Chinês, entre outros
12	Celim	IN	PN	Java	Todos	JDBC*	PC	Catalão, Espanhol, Francês, Inglês
13	CK-Ledger	IN	DC					Projeto Descontinuado
14	CK-ERP	IN	BT	PHP	POSIX	MySQL Postgresql	PC	Chinês Tradicional, Chinês simplificado, Inglês
15	CK-ERP Contribs	NC	NC	NC	NC	NC	NC	NC
16	CorbaERP	IN	PN	C++ Java Phyton	Todos	Não Informado	PC	Russo, Ucraniano

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
17	Decigen	IN	PA	PHP Pearl	Todos	MySQL	PC	Inglês
18	De facto	NC	PN	Java, JavaScript	Windows, POSIX	NC	PC	Inglês
19	DotNetFinanciais	IN	PN	ASP, C#, Visual Basic	Windows	Não Informado	PC	Inglês
20	Easy ERP	NC	NC	NC	NC	NC	NC	NC
21	Edgar	NC	PN	C++	POSIX	Postgresql	NC	NC
22	edIT-pc HUB/CRM/ERP/BIS	NC	PN	JavaScriptPHP	Todos	MySQL	PC	NC
23	Enigma ERP	NC	NC	NC	NC	NC	NC	NC
24	Enterprise CRM and Groupware System	IN	BT	PHP	Todos	NC	PC	Inglês
25	Enterprise Unite Management System	IN	NC	NC	NC	NC	NC	NC
26	Erp/crm for the 3 rd world	IN	NC	PHP	NC	MySQL	PC	NC
27	ERP5							
28	ERP para la Industria Metal-mecânica	IN	PN	Java	Todos	Não Informado	PC	Espanhol
29	ERP-system for small enterprises	ED	PN	Java	Todos	Oracle 8i	PC	Inglês, Russo
30	ERPBr	IN	PN	JavaScript / PHP/ Unix Shell	POSIX	SQL Server MySQL Oracle PostgreSQL Sybase	PC	Português (Brasil)
31	erpFramework	NC	NC	NC	NC	NC	NC	NC
32	ERP Livre	NC	NC	NC	NC	NC	NC	NC
33	Erp-php	IN	BT	PHP	Posix	NC	PC	Espanhol
34	ERPmyne	IN	BT	Java	Todos	MySQL	PC	Espanhol
35	Evaristo	IN	PS	Java	Todos	JDBC*	PC	Português (Portugal)
36	EzyBiz	IN	PS	PHP	Todos	MySQL	PC	Inglês
37	Factura LUX	IN	PS	C++	Linux, Windows,MAC		PC	Espanhol Português
38	Free ERP	IN	NC	PHP	NC	MySQL	NC	NC
39	FreeCRM	NC	PA	PHP	Todos	Não Informado	PC	Inglês
40	Freedom ERP	IN	PS	Java	Todos	Firebird, Interbase, JDBC	PC	Português (Brasil)

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
41	Gardenia	NC	BT	Java, JavaScript, JSP	Todos	JDBC, Firebird, Interbase	PC	Português
42	GECO GEstione e COntabilità Open Source	CO	NC	PHP	Windows, Linux	NC	PC	Italiano
43	Gemma	IN	PN	Ruby	Windows, Linux	MySQL, SQLite	PC	Inglês e Italiano
44	General ERP	NC	PN	Delphi / Kylix, Java, Javascript, PHP	Todos	NC	PC	Chines simplificado e tradicional, Ingles
45	Gest Management	NC	PN	JavaScript PHP	Windows, POSIX	NC	PC	Italiano
46	Gibbon	CO	AP	Perl	POSIX	NC	PC	Ingles
47	Global Operating System	NC	PN	Assembly, C, C++, Java, PHP	Todos	NC	NC	Chinês (Simplificado), Inglês, Francês, Alemão, Italiano, Espanhol
48	Global J2EE Manufacturing Solution	IN	NC	Java	Todos	Oracle MySQL Access SQL Server WebLogic JBoss	PC	NC
49	Gnuerp	IN	PN	C, PHP	Windows, POSIX	MySQL	PC	Inglês
50	Gerencial	IN	PN	C++, Delphi, Kylix, Java, PHP	Windows, POSIX	Firebird, Oracle, SQL Server, Interbase	PC	Português (Brasil)
51	Gerp - Gerenciador de Processos	NC	NC	PHP	NC	NC	NC	Português (Brasil)
52	Harmony	IN	PN	Java	Todos	NC	PC	Ingles
53	Humano2							
54	ICPAC	IN	AP	C#	Todos	NC	PC	Ingles, Espanhol
55	InterHost Advanced Hosting Solution	SE	PN	C++, Java JavaScript, Perl, PHP, Unix Shell	POSIX	NC	PC	Ingles
56	ipanema	NC	NC	Java	NC	NC	NC	NC
57	Iseries ERP	IN	PN	NC	IBM	NC	IBM	Inglês

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
58	IpyME	IN	PN	PHP	Todos	MySQL, Postgresql	PC	Espanhol
59	Java Administration Components	NC	PN	Java PL/SQL	NC	NC	NC	Inglês
60	Jazz ERP	IN	PN	Java PL/SQL	Todos	JDBC	PC	Inglês
61	JBasics -- Common EJB components	SE	PN	Java, JavaScript	Todos	NC	NC	Inglês
62	Jerpa	NC	PA	JAVA	Todos	NC	PC	Alemão, Inglês
63	JewelERP	IN	PN	Java	Windows, POSIX	NC	PC	Alemão, Inglês
64	Jforest	NC	NC	Java	Todos	MySQL	PC	NC
65	Kalpa	IN	PA	C++	Linux, Windows	Postgresql	PC	Inglês, Russo
66	Keko	NC	PN	C++	POSIX	NC	PC	Inglês
67	Kukay	IN	PA	Java	Todos	NC	PC	Inglês, Turco
68	Komombo	IN	PA	ASP, C#	Windows, POSIX	MySQL	PC	Italiano
69	KrystalERP.Net	NC	PN	C#	WinXP, Windows Server 2003	ADODB, SQL Server, MySQL, Oracle	PC	Inglês, Francês
70	Keen CRM/ERP	IN	AP	PHP	Windows, POSIX	MySQL	PC	Italiano
71	Lafi	IN	PN	Python	Windows, POSIX	Database API	PC	Alemão, Inglês
72	Linux-Kontor	IN	PS	Java	Linux, Windows, MAC	Adabas, Postgresql	PC	Alemão, Inglês
73	LinuxStok	SE	BT	PHP	Windows, POSIX	MySQL, pgsq, SQLite	PC	Portugues
74	Linux Small-MediumBusiness ERP solutions	NC	NC	NC	POSIX	NC	NC	NC
75	MaiaERP	IN	PN	Delphi, Kylix	Windows	Firebird	PC	Espanhol
76	manageXcel	IN	PN	PHP	Todos	ADODB*	PC	NC
77	Mapuche	NC	NC	NC	NC	NC	NC	NC
78	Medical Information System	SE	NC	NC	NC	NC	NC	NC
79	Manager SQL	NC	NC	C/C++	NC	NC	NC	NC

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
80	Mercury Business Solution	IN	PN	Java	Todos	NC	PC	NC
81	Microerp ERP System	IN	SAI	Java	Todos	JDBC*	PC	NC
82	Milonga	NC	NC	PHP	NC	MySQL	PC	NC
83	Mindelo	IN	PA	PHP	Todos	NC	PC	Inglês
84	MooN	NC	PA	Java	Todos	NC	PC	Inglês
85	mTeamwork	NC	NC	NC	NC	NC	NC	NC
86	myco project, the	ED	AP	Perl	Posix	NC	PC	Inglês
87	MyOpenERP	IN	PN	Delphi, Kylix, Java, PHP	Todos	NC	PC	Alemão
88	Neogia	IN	BT	Java	Todos	JDBC*	PC /	Francês, Inglês, Italiano, Português (Brasil)
89	no-X Business Suite	NC	NC	PHP	NC	MySQL	NC	NC
90	OAJ (OpenAccountingJ)	IN	PA	Java	Todos	MySQL	PC	Inglês
91	Open Business Solution	IN	PA	Java	Window, POSIX	NC	PC	Inglês
92	Open Business Suite	CO	PN	PHP, Perl	Todos	MySQL	PC	Inglês, Francês
93	OpenAspect	IN	BT	C++, Java, PHP	Todos	Firebird, Interbase	PC	Inglês
94	OpenERP	NC	NC	PHP, Python	NC	NC	NC	NC
95	Open Enterprise Management Suite	IN	PA	PHP	Todos	Pear:DB	PC	Alemão, Inglês
96	Open j2ee ERP Project	NC	NC	Java	NC	NC	NC	NC
97	Openlogistics	IN	PA	Java	Todos	NC	PC	Inglês
98	OpenSchool ERP	ED	PN	Java, JavaScript, PHP	Todos	NC	NC	Inglês, Hindu, Telugu
99	OpenImovel	SE	PA	Python	Todos	MySQL	PC	Português (Brasil)
100	OpenTASSEL	ED	AP	Java JavaScript	Windows, POSIX	Oracle, Postgresql	PC	Inglês

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
101	openXpertya	IN	PS	Java, JavaScript	Todos	JDBC, Firebird, Interbase, Oracle, Postgresql, Sybase	PC	Português (Brasil), Catalão, Galícia, Português, Espanhol
102	open shoe managment system	NC	PA	PHP	Todos	MySQL	PC	Ingles, Alemão
103	Optres	NC	NC	NC	NC	NC	NC	NC
104	Oratio	NC	PS	Perl, PL/SQL	Window, POSIX	NC	NC	Ingles, Frances, Grego, Italiano, Espanhol
105	orderline24	NC	NC	NC	NC	Access, MySQL, MyODBC	NC	NC
106	OSSUITE	IN	AP	PHP	Todos	Não informado	NC	Inglês
107	Oxerp	IN	BT	Java	Todos	MySQL, Postgresql	PC	Inglês, Francês
108	Palmtree	NC	NC	NC	NC	NC	NC	NC
109	PECAP	NC	NC	NC	NC	NC	NC	NC
110	phpCompany	IN	PN	PHP, Perl	Todos	Não Informado	PC	Inglês
111	phpERP	NC	PN	PHP	POSIX	NC	PC	Inglês
112	phpMyERP	IN	PA	PHP	Todos	MySQL	PC	Espanhol, Inglês
113	PHPSME	IN	NC	NC	NC	NC	NC	NC
114	PHPKernel	NC	PN	PHP	Todos	NC	PC	Inglês
115	Plazma	SE	PN	Java	Windows, POSIX	JDBC	PC	Ingles, Russo, Ucraniano
116	ProductTycoon							
117	PROKARTEL	NC	NC	NC	NC	NC	NC	NC
118	PucERP	CO	NC	Java	NC	NC	NC	NC
119	PyMEsLibres	IN	AP	C#	Windows, POSIX	MySQL, Postgresql	PC	Espanhol, Inglês
120	PyERP	IN	PN	Python	Windows, POSIX	MySQL	PC	Inglês, Português
121	PyPYME	IN	PN	Python	Todos	NC	PC	Espanhol
122	PXS ERP/CRM	NC	NC	NC	NC	NC	NC	NC
123	QMarket	NC	NC	PHP	NC	NC	NC	NC
124	QIQ ERP	IN	PN	Visual Basic, ASP	Windows	NC	PC	Inglês
125	RepTux	SE	PA	Object Pascal	POSIX	Firebird	PC	Espanhol, Inglês

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
126	SAO2000 ERP/CRM	NC	AP	Java	Todos	NC	NC	Inglês, Alemão
127	Salento	NC	NC	Java	NC	Postgresql	NC	NC
128	SALU	NC	NC	NC	NC	NC	NC	NC
129	Sequoia ERP	NC	PS	Java	Windows, POSIX	DB2, SQL Server, MySQL, Oracle, Postgresql	PC	Chines (Simplificado), Ingles, Frances , Alemao , Italiano, Espanhol
130	ServiceERP	SE	NC	NC	NC	NC	NC	NC
131	Shalma	NC	PA	Java	Todos	Postgresql	PC	NC
132	SIA – Sistema Integral de Administration	CO	PA	Java	Todos	MySQL	PC	Espanhol
133	Sistema de Gestão para Grupos Escoteiros	ED	PN	PHP PL/SQL	Todos	NC	PC	Português (Brasil)
134	SPA – Sítio do Pica-pau Amarelo	IN	PA	NC	Windows, POSIX	NC	PC	Português
135	SoopERP	IN	PN	PHP	Todos	MySQL	PC	NC
136	Soul	NC	PN	Java	Todos	NC	NC	Português (Brasil), alemao, holandes, ingles
137	Suniant ERP	IN	BT	PHP, Unix Shell	Windows, POSIX	Postgresql	PC	Chinês (Tradicional e Simplificado), Francês
138	Takas ERP & CRM	NC	PN	Python	Todos	NC	PC	Inglês, Hungaro
139	Tangrim ERP	NC	NC	NC	NC	NC	NC	NC
140	TEAM MRPII System	IN	NC	NC	NC	NC	NC	NC
141	Tiny ERP	NC	PS	Python XSL	Todos	SQL-based	PC	Alemão, Inglês, Francês, Espanhol
142	Transportation ERP System	NC	NC	NC	NC	NC	AS/400	NC

	Nome Sistema	Área aplicação	Estágio do projeto	Lingua-gem de programação	Sistema Operacional	Banco de dados	Hardware	Idioma
143	TUTOS	NC	PN	PHP	Todos	Firebird, Interbase, MySQL, Oracle, Postgresql	PC	Português (brasileiro), Catalão, Chinês (Simplificado), Chinês (Tradicional), Alemão, Inglês, Francês, Holandês, Grego, Húngaro, Italiano, Japonês, coreano, Polonês, Russo, Espanhol
144	UCSG PROG IV ERP	SE	PN	Java	Todos	MySQL	PC	Não informado
145	Uniform ERP	SE	PN	C++	POSIX	NC	PC	Inglês
146	University ERP System	ED	PS	Java	Todos	MySQL, Oracle	PC	Inglês, Malaio
147	Value	IN	OS	Delphi/ Kylix, Java	Windows, POSIX	MySQL, Oracle, Postgresql	PC	Inglês
148	Vianobis	NC	PA	Java	Windows, POSIX	NC	PC	Inglês, Português
149	Visual ERP	NC	PN	ASP, Visual Basic	Windows	Todos	PC	Inglês
150	XERP	IN	AP	Java	Todos	JDBC*	PC	Espanhol,
151	XTECH							
152	web based instant messenger	NC	PA	C++ Java JavaScript	Todos	NC	PC	Inglês, Japonês
153	WebERP	IN	OS	PHP	Todos	MySQL, Postgresql	PC	Espanhol, Inglês, Polonês, Português (Brasil), Russo
154	WebPortal-Intranet System	NC	BT	C++, Delphi/ Kylix, PHP	Todos	MySQL	PC	Inglês, Húngaro, Romeno
155	WEBSAFI	IN	PA	PHP	POSIX	NC	PC	Espanhol
156	We-Skey	IN	PA	PHP	Todos	MySQL	PC	Português (Brasil), Inglês, Italiano
157	Witima ERP, The	CO	AP	PHP	Todos	NC	PC	Inglês

	Nome Sistema	Área aplicação	Está-gio do projeto	Lingua-gem de programação	Sistema Operacio-nal	Banco de dados	Hardware	Idioma
158	Wyatt	IN	PN	Java	Todos	NC	PC	NC
159	WyattERP	NC	PS	C, Tcl	POSIX	Postgresql	PC	Ingles
160	Yuza Open Erp	NC	BT	C#	Windows	ODBC, Access, SQL Server, MySQL, Oracle	PC	Italiano
161	ZERP							
162	Zion-IT-ERP	NC	PA	PHP	Todos	MySQL	PC	Inglês

ANEXO A – RELAÇÃO DE LICENÇAS DE *SOFTWARE* LIVRE

Licenças de *Software* Livre compatível com a GPL:

- A Licença Pública Geral GNU, ou GNU GPL.
- A Licença Pública Menos Geral do GNU, ou GNU LGPL.
- A Licença do Guile
- A Licença do X11.
- A Licença do Expat ou Licença do MIT
- Standard ML of New Jersey Copyright License
- A Licença Geral do Cryptix.
- A licença modificada do BSD.
- A licença da Zlib.
- A licença da Biblioteca Padrão de Funções do iMatix.
- A Licença e Nota de *Software* do w3c.
- A Licença de Banco de Dados de Berkeley (conforme publicada em 12/09/1999).
- A Licença do Python 1.6 a 2 e versões anteriores
- A Licença do Python 2.0.1, 2.1.1, e versões posteriores
- A licença do Perl.
- A licença do Javascript da Netscape

Licenças de *Software* Livre incompatível com a GPL:

- A Licença Pública Arphic.
- A Licença Original do BSD.
- A Licença do Apache, Versão 1.0.
- A Licença do Apache, Versão 1.1.
- A Licença Pública do Zope.
- A Licença do xinetd.
- A Licença do Python 1.6b1 e versões posteriores.
- A Licença do OpenLDAP, Versão 2.3.
- Licença Pública da IBM, versão 1.0
- A Licença do Phorum

- A Licença Pública do Mozilla (MPL).
- A Licença de Código Aberto do Netizen (NOSL), versão 1.0.
- A Licença Pública do Interbase (IPL), versão 1.0.
- A Licença Pública da Sun.
- The Nokia Open *Source* License.
- A Licença Pública da Netscape (NPL)
- A Licença de Código Aberto do Jabber, versão 1.0
- A Licença de Código de Padrões da Indústria da Sun 1.0
- A Licença Pública do Qt (QPL).
- A Licença do *FreeType*
- A Licença de Compatibilidade Aberta
- A Licença do PHP, Versão 2.02.

Licenças que não são de *software* livre:

- A Licença Artística (Original).
- A Licença Pública de Código-Fonte da Apple (APSL).
- A Licença Comunitária de Código-Fonte da Sun.
- A Licença do Plan 9
- Licença Pública Aberta
- A Licença de Código-Fonte do Solaris (*Foundation Release*), versão 1.1
- A Licença do YaST
- As Licenças de Daniel Bernstein