

Complexo eletrônico: introdução ao software

Regina Maria Vinhais Gutierrez
Patrícia Vieira Machado Alexandre

<http://www.bndes.gov.br/bibliotecadigital>

COMPLEXO ELETRÔNICO: INTRODUÇÃO AO SOFTWARE

Regina Maria Vinhais Gutierrez
Patrícia Vieira Machado Alexandre*

** Respectivamente, gerente e economista do Departamento da Indústria Eletrônica do BNDES. As autoras agradecem a colaboração da estagiária de engenharia Camila Pinto Caldeira, do coordenador de serviços Arthur Adolfo Guarido Garbayo do Centro de Pesquisa de Informações e Dados do BNDES, dos analistas de sistemas Alexandre da Silva Pereira e Gladstone Moisés Arantes Junior, das empresas Microsiga e Motorola, do CESAR, bem como dos entrevistados José Eduardo Roselino, Giancarlo Nuti Stefanuto, Paulo Roberto Tosta da Silva, Ângela Maria Alves e Silvia Frick.*

COMPLEXO ELETRÔNICO

Resumo

Sendo o principal insumo do software a inteligência humana, este setor é um grande absorvedor de mão-de-obra qualificada. Sua importância, porém, não se restringe a esse fato. O poder transformador do software estende-se bem além das fronteiras do complexo eletrônico, atingindo praticamente todos os setores da atividade humana.

Através do estudo da história do setor, este trabalho busca as raízes da configuração atual das forças de mercado atuantes no mundo e, por extensão, no mercado brasileiro. Analisa também o surgimento do software livre, suas potencialidades e sua incorporação às estratégias empresariais da indústria.

Introdução

O poder transformador do *software*¹ estende-se bem além das fronteiras do complexo eletrônico, atingindo praticamente todos os setores da atividade humana. A automatização de tarefas repetitivas, o aumento de controle e eficiência dos processos, a possibilidade de antecipação de problemas e sua solução prévia – caso das simulações – são apenas algumas das possíveis aplicações dessa tecnologia.

A análise da trajetória do *software*, desenrolada em paralelo à própria história da informática, embora a ela não mais se restrinja, permite constatar a presença maciça do Estado americano no nascedouro dessa indústria e em cada passo importante que ela tem dado. E não apenas através de compras públicas, mas principalmente através de encomendas de novos projetos a universidades e empresas.

Do uso militar, o conhecimento tem transbordado para aplicações comerciais, o que justifica o amplo domínio dos Estados Unidos nessa indústria, pois a maior parte das grandes corporações internacionais do *software* lá estão baseadas. O mercado norte-americano é também o seu principal mercado, conferindo a essas empresas uma escala doméstica que viabiliza amplamente a sua atuação internacional.

O *software* tem passado por profundas modificações, seguindo o aumento da capacidade de processamento e de memória das máquinas, e está presente também em todos os produtos eletrônicos, enriquecendo e transformando a sua funcionalidade. Além disso, junto com a disseminação da eletrônica embarcada por outras cadeias produtivas, é um dos responsáveis pela diferenciação de produtos tradicionais, conferindo-lhes maior competitividade.

A multiplicidade de usos e produtos de *software* é enorme, assim como é intenso o dinamismo dessa indústria. A todo momento surgem novas oportunidades, configuradas por aplicações e funcionalidades inéditas, ao mesmo tempo em que *softwares* já consagrados estendem seu uso em novos mercados, consolidando posições.

Multiplicaram-se as camadas de *software*, em protocolos de comunicação variados, que permitem a conexão de equipamentos, e usuários através deles, por todo o mundo. O melhor ícone dessa integração é a internet, a grande rede mundial para a qual afluem redes regionais, locais e privadas. Porém, subjacente a essa multiplicidade, existe um grande esforço de desenvolvimento – basta

¹ Conjunto de instruções e dados que permitem a um computador a realização de tarefas previamente programadas, seja por uma máquina (hardware) ou outros softwares.

lembrar que o principal insumo da indústria de *software* é o conhecimento humano, o presente e o acumulado.

Por ser um bem intangível, resultado de produção intelectual, considerou-se adequada a proteção do *software* por legislações baseadas em propriedade intelectual. Entretanto, nem sempre foi assim. No início da informática, o *software* era apenas um apêndice do *hardware*, com o qual era fornecido, até porque esse *software* era pequeno e fortemente dependente do *hardware* específico para o qual havia sido construído.

Tendo a internet como grande exemplo de desenvolvimento comunitário, surgiu nos Estados Unidos, em 1984, o movimento do *software* livre, que, em oposição à propriedade intelectual, criou para o *software* uma nova forma de licenciamento.

O movimento foi, aos poucos, espalhando-se por todo o mundo. Ao final da década de 1990, o *software* livre foi reconhecido pelas grandes multinacionais da indústria, que passaram a adotá-lo como parte das soluções proprietárias vendidas. Contudo, é importante observar que esse é um campo em que o debate jurídico apenas começou.

Seja direcionado à solução de questões sociais como o *apartheid* digital ou visando ao fim do uso não autorizado de *software* que grassa largamente nos países emergentes, a opção pelo *software* livre externalizada por diversos governos no mundo tem trazido uma nova frente a esse embate.

No Brasil, a indústria autóctone de *software* é muito jovem, com pouco mais de 10 anos de idade, e constituída em grande maioria por empresas de porte médio e pequeno. Essas empresas oscilam entre a prestação de serviços, normalmente gravitando ao redor dos produtos importados das grandes corporações internacionais, e o desenvolvimento de produtos para aplicações específicas, como, por exemplo, a financeira.

Como fortalecer essa indústria, de forma a aumentar sua participação no mercado interno – hoje muito inferior a 50% –, e torná-la competitiva internacionalmente são os desafios colocados.

Foi visando conhecer melhor a indústria de *software* brasileira, no sentido de identificar as forças que agem nesse mercado e construir novas formas de atuação no setor, que o BNDES realizou o estudo ora apresentado.

Classificação do Software

Existem várias maneiras de estudar e classificar o *software*, sendo algumas delas apresentadas nesta seção. Inicialmente, porém, será feita uma segmentação baseada no modelo de negó-

cios, o que permite dividir o *software* em três categorias (produtos, serviços e embarcado), analisadas a seguir.

Tradicionalmente, os produtos de *software* têm sido divididos em três categorias, apresentadas na seqüência: infra-estrutura, ferramentas e aplicativos. Cabe observar que essas categorias e as respectivas segmentações estão em evolução. Novos produtos são lançados incorporando funcionalidades de outros, ao mesmo tempo em que a evolução de um dado produto leva à expansão de suas funcionalidades, o que torna as fronteiras entre os segmentos, em alguns casos, indefinida.

Produtos de Software

Classificação Técnica

Infra-Estrutura

Sistema Operacional

É o primeiro e mais importante componente do *software* de infra-estrutura, sendo responsável pelo controle e acesso aos recursos do *hardware* do sistema – processador, memória, interfaces de periféricos – e pela realização de funções básicas, permitindo a execução de todos os outros *softwares*, como os aplicativos. É conhecido, coloquialmente, como o *software* que acompanha a máquina, sendo ele que permite que outros *softwares* sejam instalados.

Seu núcleo e módulo principal chama-se *kernel*, o qual provê acesso seguro ao *hardware* por diferentes programas, que disputam tais recursos simultaneamente, cabendo ao *kernel* gerenciar o acesso ao recurso pretendido. O *kernel* permite ainda que os programas aplicativos relacionem-se com o sistema operacional de maneira uniforme, independentemente do *hardware* subjacente. Isso é feito através de abstrações do *hardware*, ou seja, um conjunto de rotinas que emulam detalhes específicos das plataformas, ocultando a sua complexidade dos programadores e deixando para o *kernel* a tarefa de ajustar-se às especificidades da máquina.

Além do *kernel*, outros módulos compõem um sistema operacional, como, por exemplo, o gerenciador de arquivos, que facilita a criação e a manutenção de diretórios e arquivos. As características dos sistemas operacionais variam enormemente, como nos casos citados a seguir:

- Alguns sistemas executam apenas um programa aplicativo de cada vez e são qualificados como monotarefa. Em oposição a estes, têm-se os sistemas multitarefas, que permitem que vários aplicativos “rodem” concorrentemente. Na verdade, todos os programas são carregados na memória da máquina, porém só um deles, a cada momento, está ativo, havendo uma rápida alternân-

cia do controle entre eles (concorrência), o que dá a impressão de simultaneidade.

- Há sistemas que possuem apenas um usuário, sendo por isso chamados de monousuários. De forma inversa, existem os multiusuários, que possibilitam a vários usuários compartilharem um mesmo computador, local ou remotamente.
- De forma análoga, há os sistemas monoprocessados, com apenas uma unidade de processamento (CPU), e os multiprocessados, com mais de uma CPU.

Os sistemas operacionais mais recentes possuem também módulos de comunicação em rede, prevendo, por exemplo, a utilização de protocolos TCP/IP (*transmission control protocol/internet protocol*).

É possível que um programa seja executado diretamente sobre a máquina, sem o uso de sistema operacional. Contudo, isso aumenta em muito a sua complexidade, exigindo conhecimentos extensos sobre o *hardware* que está sendo usado e requerendo que o computador seja inicializado e recarregado a cada novo programa, como era o caso dos primeiros computadores.

Programas Servidores

Um outro tipo de *software* de infra-estrutura bem conhecido é o programa servidor, camada executada sobre o sistema operacional e que está associado à realização de uma tarefa específica, como, por exemplo, o servidor de correio eletrônico.

Middleware

Camada de *software* que liga duas ou mais aplicações em plataformas distintas, conectando-as através de diferentes redes de comunicação. Provê segurança de acesso e troca de informações, permitindo a integração de diferentes plataformas e sistemas legados (preexistentes) para o uso transparente e distribuído de seus recursos.

Gerenciador de Redes

Software voltado para o gerenciamento de redes de comunicação, permite gerenciar a topologia da rede, provendo e reconfigurando os seus elementos. Possibilita também registrar o uso de recursos e diagnosticar e solucionar falhas, medindo e otimizando o desempenho da rede.

Gerenciador de Armazenagem

Camada de *software* sobre o sistema operacional que possibilita a criação de uma infra-estrutura de armazenagem de dados,

em geral separada da máquina hospedeira do sistema operacional. Permite o gerenciamento do *hardware* dedicado à armazenagem, com o acompanhamento de seu estado e da capacidade dessa armazenagem, bem como o controle da tendência de crescimento dos dados armazenados, visando equilibrar a disponibilização de serviços adequados aos usuários e a redução dos custos.

Gerenciador de Sistemas

É um *software* que possibilita o gerenciamento de sistemas – *hardware* e *software* –, de forma a reduzir o tempo e o custo de operações necessárias ao acompanhamento de rede de computadores e de comunicação. Permite que sejam feitos remotamente diagnósticos e identificação de falhas, inventários de *hardware* e *software* e distribuição e desinstalação de *software*. O uso de *software* pode ser rastreado e medido por usuário, grupo, tempo ou quota, de modo a possibilitar a forma de pagamento mais adequada por licenças de uso. Já a distribuição de *software* pode ser endereçada a máquinas, usuários, grupos ou segmentos de rede.

Segurança

Essa categoria compreende todos os tipos de *software* necessários ao controle e monitoramento do acesso a recursos internos e externos de uma empresa, o que inclui: a) antivírus; b) filtros de conteúdo para internet como *anti-spam*;² c) *firewalls*;³ d) produtos para detecção, identificação e isolamento de tentativas de acesso não autorizado a recursos do sistema; e) *software* de criptografia; f) sistemas para administração de segurança, englobando dispositivos de *hardware* e *software*; etc.

Ferramentas

Conjuntos de programas que auxiliam a construção de outros programas e aplicações.

Linguagens de Programação

Uma das mais antigas ferramentas são os *softwares* de linguagens, as quais têm por objetivo a construção de programas em código de alto nível, isto é, próximo à linguagem humana ou utilizando símbolos de fácil compreensão, e a sua transformação em código binário – formado por seqüências de zeros e uns – ou executável. O código executável é assim chamado pelo fato de somente a linguagem binária ser “entendida” pela máquina. Alguns dos principais programas que compõem um *software* de linguagem são apresentados a seguir.

O editor é o programa que permite compor ou modificar outros programas, criando e alterando textos de instruções através

²Spam é o envio, pela internet, de múltiplas cópias da mesma mensagem, normalmente divulgação comercial de negócios ou serviços de legalidade duvidosa, a destinatários que não desejariam recebê-las.

³Recebem todo tipo de mensagens pela internet, permitindo ou não que seja feita a conexão, e visam proteger o sistema particular de ataques, sejam as ameaças reais ou potenciais, assegurando que nenhuma mensagem não autorizada entre ou saia do sistema.

de ações como incluir, apagar, trocar trechos de lugar (cortar e colar) etc., e dá origem ao que se chama código-fonte de um programa.

O compilador traduz o código-fonte em código binário ou executável. Normalmente, cada instrução do código-fonte dá origem a um conjunto de instruções binárias. Essa “explosão” é feita pelo compilador, que também verifica a ocorrência de erros de sintaxe no código-fonte e otimiza o código binário gerado, de forma a tornar a sua execução mais eficiente.

Em seguida, o código binário gerado pelo compilador é ligado a funções e procedimentos – rotinas, de forma geral – padrões da linguagem, predefinidos em bibliotecas já existentes. O código binário assim “linkado” (conectado a componentes de bibliotecas) recebe o nome de código executável.

É bem conhecido também o debugger, utilizado para testes e análises de eficiência, assim como o construtor de interfaces de uso gráfico (GUI), o IDE.

Existem ainda outros programas de linguagem, como os montadores e os interpretadores. Os montadores (*assemblers*) foram as primeiras linguagens utilizadas, fazendo a tradução direta de instruções em linguagem próxima da humana para a linguagem de máquina, o que requer um profundo conhecimento do funcionamento dessa máquina.

Muitas são as linguagens de programação criadas em diferentes gerações, sendo as mais recentes de mais alto nível que as da geração anterior. Algumas das mais conhecidas são: Fortran (*formula translator*), criada para aplicações científicas e de engenharia; Cobol (*common business oriented language*), voltada para aplicações comerciais e corporativas; Pascal, criada para fins didáticos e adotada em aplicações para microcomputadores; C e C++, Delphi, Visual Basic, Java e .Net, voltadas para o desenvolvimento em geral e largamente adotadas pela indústria de *software*; etc.

Com a evolução do *software*, muitos outros tipos de ferramentas surgiram, sendo os principais apresentados a seguir.

Gerenciamento de Desenvolvimento

Software voltado para suportar a análise e o projeto de aplicações. Existem ambientes de desenvolvimento que permitem a realização de diversas atividades, como, por exemplo: modelagem de negócios, processos e dados; gerenciamento de requisitos (especificações); gerenciamento de versões e alterações; modelagem de arquitetura e projeto; geração automática de código; teste de componentes e análise de atividades em tempo de execução; acompanhamento da medição de qualidade do *software*; etc.

Modelagem de Dados

Software que permite a organização de dados e processos, através da definição de formatos e estruturas, e a construção de bancos de dados particularizados para aplicações, definindo as respectivas ações de pesquisa, acesso, recuperação e descarte de dados. Essa classe engloba diversos tipos de tecnologias mais ou menos recentes – pré-relacionais, relacionais, orientados a objetos etc.

Business Intelligence (BI)

Software que provê sistemas de informação executivos – com ferramentas para a realização de consultas e emissão de relatórios – e sistemas de suporte a decisão, baseados em dados armazenados, preferencialmente, em um *data warehouse*. Permite o acesso e a análise desses dados através de ferramentas Olap (*on-line analytical processing*), utilizando funções de modelagem e de análise estatística e apresentando graficamente os resultados. Suas interfaces são adaptadas aos usuários finais e organizadas por aspectos específicos do negócio, permitindo a realização de análises e novos planejamentos frente a metas.

Data Warehouse

Banco de dados normalmente voltado para o suporte de sistemas de informação executivos, incorpora ferramentas que permitem a extração, transformação e carga de dados originados de múltiplas fontes, geralmente operacionais – bancos de dados de várias tecnologias, planilhas etc. –, e em diferentes formatos. Possui também ferramentas que, através do exame de grande massa de dados, possibilitam a descoberta de correlações significativas, padrões e tendências, e verificam a ocorrência de inconsistências.

Ferramentas de Internet

Software destinado a suportar o desenvolvimento de aplicações para a internet, possibilitando o projeto e o gerenciamento de um *site*, através do uso de ferramentas como editores HTML, utilitários gráficos, mídia *player*, *plug-ins* etc.

Aplicativos

Softwares especializados destinados à execução de uma determinada tarefa. Frequentemente, suas entradas e saídas estão associadas a atividades humanas. Daí, a grande importância assumida pelas interfaces com os usuários no que tange à comunicação amigável e à capacidade de interação, ou seja, à sua usabilidade.

São inúmeros os aplicativos possíveis, mesmo dentro de um determinado setor de atividade, surgindo continuamente novas

funcionalidades que podem ser incorporadas a esses *softwares* ou dar origem a novos aplicativos. Como exemplo de aplicativos, têm-se os editores de textos, as planilhas, os editores gráficos, bem como os *softwares* enunciados abaixo, associados a processos de negócios.

Enterprise Resource Planning (ERP)

É um aplicativo de caráter estratégico para uma empresa por requerer profundas alterações em sua sistemática operacional, automatizando e fundamentando principalmente os processos administrativos, financeiros, de controle de ativos e de fabricação. Todas essas são aplicações de *back-office*, ou seja, não envolvem interação direta com os clientes da empresa. A construção do ERP é modular, possibilitando que essa modularidade se estenda à sua implantação.

Customer Relationship Management (CRM)

Software estratégico que visa possibilitar maior conhecimento sobre os compradores e propiciar a integração dos canais de acesso a eles com as funções de *back-office* da empresa. Para o atendimento desses objetivos, o CRM abrange atuações em três segmentos: vendas, *marketing* e serviços aos usuários.

Recursos Humanos

Conjunto de aplicativos voltados para a aquisição, o gerenciamento e a otimização da força de trabalho de uma empresa, compreendendo módulos para recrutamento, educação e treinamento, administração de pessoal, desenvolvimento organizacional, gestão de desempenho, folha de pagamento etc.

Supply Chain Management (SCM)

É também um *software* de caráter estratégico, cuja finalidade é otimizar o fluxo de produtos, serviços e informações dos fornecedores de uma empresa aos seus clientes. Normalmente, é dividido em módulos voltados para o planejamento das operações de manufatura, a execução das operações planejadas (desde a emissão de ordens de compra até a entrega aos clientes) e o gerenciamento das operações de suprimento.

Todos esses são *softwares* de uso geral, havendo produtos específicos por setores de aplicação, como *software* para controle de rebanhos, apoio à correção de solo, tarifação de serviços de comunicação, educativos etc.

Existem ainda alguns *softwares* bastante especializados, como, por exemplo, aqueles voltados para o desenvolvimento de projetos de engenharia – de bens mecânicos a circuitos integrados ou de apoio à manufatura.

Uma forma tradicional de classificar o produto de *software* toma por base o mercado ao qual se destina, o que permite distingui-lo como *software* horizontal ou vertical.

Classificação quanto à Inserção no Mercado

Horizontal

O *software* horizontal é aquele que, a princípio, pode ser utilizado por qualquer tipo de usuário, sendo necessários para a sua construção apenas conhecimentos de informática. Figuram nessa categoria todos os *softwares* de infra-estrutura, ferramentas e aplicativos genéricos como editores de texto, planilhas, editores gráficos, agendas etc.

Vertical

Já o *software* vertical está relacionado ao tipo de usuário ou de atividade por ele desenvolvida. Sua construção requer, além de conhecimentos de informática, conhecimentos específicos da atividade ou negócio do usuário. São exemplos de *software* vertical os sistemas para administração hospitalar, projeto de circuitos integrados, previsões meteorológicas, os *softwares* educativos etc.

É comum que os produtos de *software* sejam classificados quanto à forma de comercialização, dividindo-se entre as categorias padronizado ou pacote, customizado e sob encomenda.

Classificação quanto à Forma de Comercialização

Pacote

Produtos padronizados ou pacotes são aqueles totalmente desenvolvidos antes do seu lançamento no mercado. Obedecem a uma especificação padrão, que busca atender às necessidades médias dos usuários incorporando também algumas funcionalidades específicas, de forma que, apesar de o uso de todas as funcionalidades por um mesmo usuário ser improvável, a maioria dos usuários sintam-se atendida. É fraca a relação entre empresa desenvolvedora e usuário. Pertencem a essa categoria os *softwares* de infra-estrutura, ferramentas e alguns aplicativos.

Customizado

Produtos customizados são aqueles em que a grande maioria dos seus módulos é desenvolvida previamente ao lançamento no mercado. Embora obedçam a uma especificação padrão, são feitas adaptações do produto a cada usuário ou instalação em particular. Além da fixação de parâmetros, a customização pode implicar o desenvolvimento de novas funcionalidades ou de um ou mais módulos específicos para a instalação. A relação entre desenvolve-

dora e usuário é forte, sendo consequência dessa interação a capacitação da desenvolvedora no tipo de negócio do usuário. Pertencem a essa categoria alguns *softwares* aplicativos.

Sob Encomenda

Por fim, os *softwares* sob encomenda caracterizam-se pelo fato de serem desenvolvidos para atendimento a necessidades exclusivas de um usuário. Como exemplo, poderia ser citado o sistema para processamento de declarações de ajuste do imposto de renda, cujo usuário único é a Receita Federal. Outro caso seria o de um *software* inovador, criado para dar solução a um problema específico de um usuário nunca antes resolvido por *software*. A interação entre desenvolvedora e usuário é intensa, sob pena de o *software* não corresponder às expectativas e necessidades do usuário, as quais comumente variam ao longo do desenvolvimento. A partir do desenvolvimento do novo produto, a empresa de *software* torna-se apta a criar um produto padronizado para atender a outros clientes com o mesmo tipo de problema, se eles existirem e se não houver venda da propriedade intelectual ao usuário original. De forma geral, somente aplicativos são desenvolvidos sob encomenda.

Em tese, qualquer programa pode ser desenvolvido sob encomenda, porém sua relação custo/benefício é francamente desfavorável quando já existem soluções no mercado, o que leva o usuário a optar pela padronização sempre que tal substituição seja possível e não haja razões de sigilo envolvidas.

É necessário observar aqui a diferença entre o produto desenvolvido sob encomenda, cuja especificação é feita pela desenvolvedora de *software*, e a especificação de um produto pelo próprio usuário, que, podendo optar pelo desenvolvimento do *software* em casa, prefere contratar a realização desse serviço a terceiros – uma fábrica de *software*.

Serviços de Software

Denominam-se serviços profissionais de Tecnologia da Informação (TI)⁴ as atividades tradicionais que demandam conhecimentos especificamente relacionados a essa tecnologia, compreendendo consultoria, desenvolvimento de aplicativos (*software* sob encomenda), integração, treinamento, suporte técnico e manutenção, entre outros.

Por integração compreende-se o detalhamento e o gerenciamento da implantação de um *software* ou sistema, de maneira a garantir o perfeito funcionamento do *software* e a sua interligação com os demais sistemas, *softwares* ou infra-estrutura.

⁴Expressão que compreende de todas as formas de criar, guardar, trocar e usar informação, em qualquer de suas formas, nascida da confluência entre informática e telecomunicações.

O suporte é uma atividade de apoio ao uso, incluindo a prestação de esclarecimentos sobre a operação do produto e a orientação para a sua melhor utilização, além da identificação de problemas e falhas no produto. Já a manutenção implica fazer com que o produto continue atendendo adequadamente ao usuário, através da remoção de falhas identificadas e da agregação de pequenas funcionalidades.

O crescimento e a disseminação da TI transformaram a maneira com que esses serviços são prestados e viabilizaram a realização de outros tipos de serviços, não diretamente ligados com a informática e as telecomunicações, mas que delas fazem intenso uso. Daí serem considerados de TI, como, por exemplo, o *call-center*.

Assim, atualmente pode-se melhor caracterizar os serviços de TI dividindo-os, de acordo com o método de compra, em dois grandes grupos: serviços discretos e *outsourcing*.

São aqueles realizados em um período de tempo curto e predeterminado. Os contratos de tais serviços são relativamente simples, ficando a responsabilidade do projeto com o cliente.

Incluem-se nesse grupo os serviços profissionais tradicionais. Tais serviços são buscados quando o cliente não é capaz de realizá-los internamente de forma eficiente – seja por falta de conhecimento ou de recursos humanos disponíveis –, procurando uma segunda parte mais bem capacitada.⁵

É cada vez mais comum a contratação de serviços por meio de *outsourcing*, que envolve a transferência de uma parte significativa da responsabilidade pelo gerenciamento para o provedor de serviços. A contratação de serviços a uma empresa externa à organização não configura necessariamente um *outsourcing*, pois ele requer sempre um nível razoável de troca de informação, coordenação e confiança entre ambas as partes, isto é, um maior grau de comprometimento do provedor de serviços com o cliente. As relações contratuais são de longo prazo e, muitas vezes, amarradas por metas de desempenho.

O nível de responsabilidade do provedor de serviço é variável, bem como o tipo de contrato estabelecido. Pode-se dividir o *outsourcing* em duas categorias distintas, apresentando complexidades crescentes:

Serviços Discretos

Outsourcing⁶

⁵A primeira contratação de serviços profissionais de TI data de 1954, com o primeiro computador empregado comercialmente. A GE queria automatizar alguns processos administrativos na sua nova planta de Louisville, Kentucky, e contratou a Arthur Andersen, devido à sua experiência pregressa em automatização de processos, para ajudá-la.

⁶Significa terceirização, mas ultimamente está associada à terceirização em TI.

- *outsourcing convencional*: terceirização de uma atividade específica da área de TI, seja ela na camada de infra-estrutura (*call-center*, *help desk*, gerenciamento de rede etc.), seja relacionada à gestão e manutenção de aplicativos (na maioria dos casos, busca-se a redução de custos, podendo ou não haver transferência de pessoas e ativos para o provedor do serviço); e
- *business process outsourcing* (BPO): pode ser definido como um contrato com uma organização externa para que esta assuma a responsabilidade em fornecer um processo ou função de negócio. O BPO vai além do *outsourcing* de aplicativos ou de infra-estrutura. Nesse negócio, o provedor do serviço é o primeiro responsável pelo projeto, assegurando o seu funcionamento, a eficiência da interface com as outras funções da empresa e a obtenção dos resultados desejados. Envolve, dessa forma, uma relação colaborativa e flexível entre o contratante e o provedor de serviço. A firma e seu parceiro freqüentemente definem o serviço a ser prestado conjuntamente, podendo ou não transferir pessoas e ativos para o fornecedor.

Software Embarcado

Caracteriza-se como embarcado aquele *software* que não é percebido nem tratado separadamente do produto ao qual está integrado, seja esse produto uma máquina, um equipamento ou um bem de consumo. Ele está presente em centrais telefônicas, terminais celulares, aparelhos de DVD, PDAs, autopeças, comandos numéricos computadorizados para máquinas-ferramenta etc. Assim, pode-se dizer que todo e qualquer bem de base eletrônica, ou que incorpore módulos eletrônicos de controle, carrega em si o *software* embarcado.

A crescente difusão da eletrônica por outras cadeias produtivas faz-se pela incorporação aos bens dessas cadeias de módulos eletrônicos constituídos simultânea e indivisivelmente por *hardware* e *software*. Este último vem aumentando sua presença pela transferência para o *software* da realização de algumas tarefas de comando e controle anteriormente realizadas por circuitos, bem como pela agregação de novas funções aos bens finais. É a disseminação da eletrônica uma das grandes responsáveis pela diferenciação desses bens, sendo incontestável a participação do *software* na determinação da sua competitividade.

As especificações de um *software* embarcado geralmente são muito rigorosas. É necessário que ele possua alta estabilidade e opere em tempo real, sendo freqüente a limitação de recursos de *hardware* como memória e discos magnéticos. Existe uma enorme gama de *softwares* embarcados, adaptados a uma imensa variedade de processadores, assim como de dispositivos de entrada e saída, de *mouses* e teclados a sensores e atuadores. A escala de produção

é muito variada, da mesma forma que os tipos de produtos que contêm o *software*.

São os equipamentos de telecomunicações os que mais rapidamente têm convertido funções de *hardware* em *software*, agregando novas funções possibilitadas pela convergência entre telecomunicações, informática e consumo. São exemplos a voz sobre IP e os terminais celulares que realizam funções de comunicação – voz, dados e imagens – e jogos.

Na grande maioria das vezes, são as próprias empresas que projetam o *hardware* as responsáveis pelo desenvolvimento do *software* embarcado, sendo poucos os casos em que ele é desenvolvido sob encomenda dessas mesmas empresas. Essa é uma das razões da dificuldade de serem construídas estatísticas sobre esse tipo de *software*.

O surgimento da atividade de desenvolvimento de *software* está estreitamente relacionado com a evolução da tecnologia da computação. Dessa forma, uma abordagem histórica da indústria de *software* acompanha o desenvolvimento da indústria de computadores.

Ao mesmo tempo, a evolução de outras indústrias como a microeletrônica também teve fortes impactos no seu desenvolvimento. Os avanços ocorridos nas últimas décadas – bem caracterizados pela Lei de Moore,⁷ que enunciava a duplicação da capacidade de processamento dos *chips* a cada ano – permitiram um salto no desempenho dos computadores, assim como uma enorme flexibilização desses equipamentos, alavancando a atividade de desenvolvimento de *software*.

A constituição de uma indústria de *software*, cuja dinâmica e forças são independentes da indústria de *hardware*, é relativamente recente, datando da segunda metade da década de 1960. Esse movimento de autonomia envolve duas dimensões: a técnica, na qual se estabelece uma atividade relativamente independente do desenvolvimento do equipamento, demandando uma série de conhecimentos específicos;⁸ e a econômica, em que se observa a formação de empresas dedicadas exclusivamente ao desenvolvimento e/ou comercialização de pacotes de *software*, as chamadas *independent software vendors* (ISV) [Roselino (1998)].

O papel proeminente dos Estados Unidos nessa área de conhecimento, por sua vez, localiza geograficamente o nascimento de uma indústria de *software* nesse país. As primeiras empresas do setor surgiram de demandas geradas por projetos estratégicos do

Histórico

⁷ Que não mais se verifica.

⁸ Desde então, uma série de ferramentas e uma extensa literatura técnica em relação aos métodos de desenvolvimento de *software* têm surgido, bem como certificações de avaliação de qualidade de produto e processo.

governo norte-americano, principalmente do Departamento de Defesa e da Nasa.⁹

O crescimento da indústria concentrou-se naquele país, abrigando até hoje as maiores empresas do setor. Nesse sentido, destacam-se as elevadas barreiras à entrada prevaletentes nessa indústria, em função das economias de escala e de rede.

Os principais fatos e acontecimentos relacionados à evolução da indústria de *software* serão apresentados a seguir (pelos motivos explanados, a referência são os Estados Unidos), os quais foram coletados principalmente dos trabalhos sobre a história da indústria de *software* realizados por Steinmueller (1995) e Campbell-Kelly (2003), fortemente recomendados caso o leitor tenha interesse em se aprofundar no assunto.

Origens (1940/65)

Os primeiros computadores foram desenvolvidos na década de 1940 para fins militares. Inexistia uma definição precisa a respeito do emprego econômico dessa nova tecnologia. A sua grande aplicabilidade na área de defesa concentrou seu desenvolvimento na esfera governamental, sendo financiado majoritariamente por recursos públicos.

O primeiro computador – um gigante eletromagnético, denominado Mark I, que ocupava aproximadamente 120 m³ – foi projetado pela Marinha norte-americana, em conjunto com a Universidade de Harvard e a IBM, em 1944. Dois anos mais tarde surgiu o primeiro computador eletrônico a válvula de propósito geral. O equipamento, denominado Eniac, foi desenvolvido por Eckert e Mauchly, também nos Estados Unidos [Gornick (1985)].

O Eniac apresentava velocidade de processamento superior à do computador eletromagnético e configurações mais próximas dos computadores atuais. Sua memória, todavia, era muito pequena, e a cada nova operação era necessário reconfigurar toda a sua fiação, exigindo um enorme esforço humano.

No mesmo período, surgiu o conceito de programa armazenado desenvolvido por Von Neumann, que propunha construir computadores que codificassem as instruções de forma que fosse possível armazená-las na sua memória. Assim, quando processassem o programa, os computadores buscariam as instruções diretamente na memória. O primeiro computador com programa armazenado foi o Edvac [Ramalho (2003)].

Pode-se dizer que o armazenamento de programas foi um marco divisor para a computação. A introdução de um programa

⁹O primeiro grande projeto em programação de tempo real foi o desenvolvimento do sistema de defesa aéreo (Sage) pelo Departamento de Defesa, que teve um papel preponderante no treinamento de pessoas nessa área de conhecimento.

armazenado na memória do computador que pudesse ser modificado para execução de outras funções proporcionou maior rapidez e versatilidade ao equipamento. A programação e, associada a ela, a possibilidade de reutilização desse programa tornaram os computadores de propósito geral mais vantajosos frente aos sistemas especializados, até então utilizados. Entretanto, foi apenas na década de 1960 que o uso de computadores de propósito geral superou o de sistemas especializados.

Uma segunda inovação, que teve forte impacto sobre a ciência da computação, foi a invenção do transistor em 1947.¹⁰ Entre as vantagens associadas à sua utilização, destacam-se: a) crescimento da velocidade de processamento; b) maior segurança nas operações; c) maior portabilidade dos equipamentos; e d) queda do custo em comparação com os computadores a válvula.

A utilização do programa armazenado e a criação dos computadores eletrônicos tornaram essa tecnologia acessível técnica e economicamente a grandes empresas e universidades. A década de 1950 marca, então, o início do uso dos computadores para fins comerciais, sendo o primeiro deles o Univac, criado em 1948.

Durante essa década, o programa de computador representava uma parcela muito pequena do sistema. Seu desenvolvimento e aprimoramento, entretanto, alavancariam as vendas de computadores, sendo interesse dos fabricantes de *hardware* estimular a sua produção. Alguns fabricantes de *hardware* forneciam, de forma casada, *softwares* básicos, como, por exemplo, interpretadores que traduziam a linguagem *assembly code* em códigos de máquina. Dessa forma, os *softwares* eram vistos ou como objetos sem valor intrínseco ou, na melhor das hipóteses, com algum valor, mas sem um mecanismo de mercado que pudesse auferi-lo.

Essa foi a estratégia adotada pela IBM, que se consolidou nesse período como líder mundial na produção de computadores de grande porte (*mainframes*). Tal liderança foi alcançada por um relacionamento próximo com seus maiores clientes – programadores dos departamentos de processamento de dados das grandes empresas –, grandes campanhas de *marketing*, em que eram promovidas novas utilidades comerciais para a computação, e desenvolvimento de *software*, que diminuía o custo e ampliava o conjunto de utilidades do computador IBM. A oferta casada da máquina com programas, como, por exemplo, sistema operacional, linguagem de programação e/ou banco de dados, buscava maximizar a utilidade dessa nova tecnologia, criando uma vantagem competitiva frente aos demais fabricantes de computadores.

O desenvolvimento de aplicações específicas ficava a cargo do próprio usuário, sendo realizado majoritariamente por equipes internas das grandes corporações, o que era facilitado pelo crescente

¹⁰Segundo Ramalho (2003), o primeiro computador com transistores, chamado de TX-O, foi concluído em 1956 no MIT. O grande dinamismo tecnológico da indústria de componentes eletrônicos – que se tornaram cada vez mais velozes e menores ao longo dos anos – tem sido uma importante mola propulsora para a indústria de computação.

uso de linguagens de alto nível. Pode-se dizer que o desenvolvimento de uma linguagem de alto nível para computadores IBM atrasou o crescimento de um mercado para o *software*. O aumento de produtividade proporcionado por esse tipo de linguagem reforçou a opção de desenvolvimento caseiro por parte dos usuários.¹¹

Essa estrutura, em que os produtores de *hardware* forneciam os programas básicos para o desenvolvimento de aplicativos e os usuários desenvolviam os seus próprios aplicativos, prevaleceu na origem da indústria da computação por um curto período de tempo.

Ao longo da década de 1960, começou a surgir uma série de empresas prestadoras de serviços de programação. Todavia, o mercado ainda era muito pequeno, fazendo com que, além de serviços de programação, elas desempenhassem outras atividades como processamento de serviços em *time sharing* (compartilhamento de tempo), teleprocessamento¹² e/ou gerenciamento de projetos.

Os maiores demandantes de serviços de programação no período eram o governo norte-americano, principalmente o Departamento de Defesa e a Nasa, e os fabricantes de *hardware*. O primeiro concentrava a grande maioria de projetos de sistemas de tempo real, tecnologia de ponta na época.¹³ A forte demanda governamental estimulou o crescimento dessa indústria no país, explicando a primazia norte-americana no setor.

Em particular, algumas empresas especializaram-se em atividades de prestação de serviços, as quais consistiam em realizar rotinas de processamento de dados para empresas que não dispunham de um computador, geralmente em função de seu menor porte, não possuindo recursos financeiros e/ou humanos suficientes para adquiri-lo. As firmas desse tipo foram chamadas de *bureau* de serviços, cujas receitas passaram de US\$ 125 milhões em 1960 (representando 13% do total das receitas da indústria de computação) para US\$ 1,9 bilhão em 1970 (elevando sua participação para 24%).

Estima-se que, no final da década de 1960, havia entre 1.500 e 3.000 empresas estabelecidas nos Estados Unidos, sendo a maioria de pequeno e médio portes. As maiores empresas cresceram através da concentração em nichos de mercado, desenvolvendo competências específicas de programação, além de adquirir ao longo do tempo habilidades em estimativa de custos e gerenciamento de projetos [Campbell-Kelly (2003, p. 63)].

Em suma, a década de 1960 foi marcada pela difusão dos computadores entre as corporações. Enquanto todos os empreendimentos de grande porte já tinham instalado computadores no final da década, as empresas médias começavam a adquirir as novas gerações de computadores (como o IBM 1401,¹⁴ que apresentava

¹¹Entre as primeiras dessas linguagens está o Fortran, introduzido em 1957. Alguns anos mais tarde surgiu a linguagem Cobol, cuja utilização foi superior à do Fortran. O Cobol foi desenvolvido por um comitê, patrocinado pelo Departamento de Defesa norte-americano, em função da identificação da necessidade de uma linguagem de alto nível para aplicações comerciais. Dois anos após a sua criação a IBM já oferecia o Cobol em vários de seus computadores.

¹²Oferta de computação remota, em que uma série de terminais era ligada por linhas telefônicas a um main-frame.

¹³Um dos primeiros grandes projetos de computação de tempo real civil foi a integração de todas as operações da American Airlines (reservas, emissão de tickets, check-in e gerenciamento de relatórios), realizada pela IBM em 1959. O sistema, denominado Sabre, teve seu custo estimado em US\$ 30 milhões e envolveu cerca de 200 profissionais por um período de cinco anos.

¹⁴O IBM 1401, lançado em 1960, foi vendido com uma nova linguagem de alto nível chamada RPG, cujas operações lembravam aquelas do sistema de cartão perfurado, podendo ser utilizado por indivíduos sem treinamento em outras linguagens. Seu sucesso foi enorme, tendo sido vendidas mais de 12 mil unidades.

porte e preços menores) e um número cada vez menor de corporações recorria ao *bureau* de serviços.

O modo de compra de *software* pela maioria dos usuários começou a mudar em meados da década de 1960. Até esse momento o *software* básico (sistema operacional e utilitários) era provido pelos fabricantes de *hardware*, enquanto os aplicativos eram desenvolvidos ou pela própria corporação ou, sob encomenda, por empresas de serviços de programação. Os aplicativos sob encomenda, 10 anos mais tarde, seriam majoritários apenas nas grandes corporações, enquanto empresas de pequeno e médio portes se defrontariam com a tarefa de escolher o melhor pacote disponível no mercado para determinada aplicação.

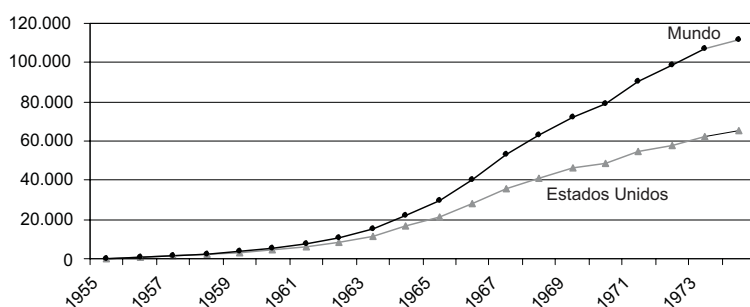
A emergência do *software* como produto pode ser atribuída a três fatores principais: a proliferação e o crescimento do poder computacional do *hardware*, a criação do IBM System 360 e a decisão da IBM em vender separadamente o *software* do *hardware* (*unbundle*).

As vendas de computadores cresceram significativamente nesse período, surgindo no mercado os de menor porte e custo, o que possibilitou sua aquisição por um número crescente de empresas. Nesse sentido, destaca-se o IBM System 360 e o minicomputador.¹⁵

Em 1960 havia 4.400 computadores nos Estados Unidos, número que já era de 21.600 em 1965, chegando a 48.500 em 1970. O Gráfico 1 apresenta a evolução das vendas de computadores nesse período [Campbell-Kelly (2003, p. 89)].

Gráfico 1

Evolução do Estoque de Computadores de Uso Geral – 1955/74



Fonte: Campbell-Kelly (2003).

O Surgimento do *Software* Pacote e sua “Produção” (1965/79)

¹⁵A Digital Equipment Corporation (DEC) foi a pioneira no mercado de minicomputadores com o lançamento do PDP-8 em 1965. O custo do aluguel mensal dessa máquina era de US\$ 525, o que representava 6% do valor do aluguel do menor modelo de mainframe da IBM (System 360/Model 30) [Steinmueller (1995, p. 18)].

Os dados, disponíveis em Steinmueller (1995), demonstram a importância do período. Os investimentos em desenvolvimento de *software* feito pelas corporações (grandes usuários) pulou de US\$ 3 bilhões a US\$ 4 bilhões em 1965 para US\$ 8 bilhões em 1970. No mesmo período, as receitas de produtos e serviços de *software* cresceram de US\$ 200 milhões para US\$ 2,5 bilhões.

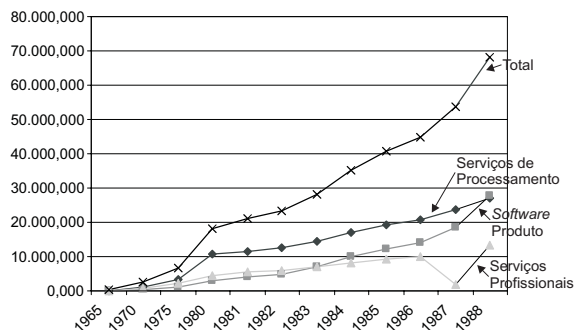
Ao mesmo tempo em que o tamanho e o preço do *hardware* diminuía, maior se tornava sua capacidade de processamento (ver Tabela 1). Acompanhando esse processo, os programas de computador tornavam-se cada vez maiores e mais complexos.

Cada vez mais recursos eram alocados nessa atividade, sendo necessário um número crescente de programadores. Com isso, uma infinidade de previsões catastróficas para a indústria começou a ser alardeada, o que ficou conhecido na literatura como a crise do *software*. A preocupação principal era de uma futura falta de programadores no mercado, dada a crescente demanda. Adicionalmente, outras questões surgiam, como a baixa produtividade dos programadores, a pequena confiança no *software* desenvolvido e o crescimento dos custos.

Gráfico 2

Evolução das Receitas Estimadas de *Software* e Serviços nos Estados Unidos segundo a Computer and Business Equipment Manufacturers Association – 1965/88

(Em US\$ Milhões)



Fonte: Steinmueller (1995).

Tabela 1

Desempenho Comparativo dos Computadores IBM – em relação ao Modelo 650 (1953)

Modelo	650	1401	360/30
Ano	1953	1959	1964
Memória (em <i>kilobytes</i>)	1	4	66
Velocidade de Processamento	1	7	43
Preço de Compra ^a	1	não disponível	0,025

Fonte: Campbell-Kelly (2003).

^aEm instrução executada por segundo.

A criação de pacotes de *software* foi a alternativa encontrada pelos fabricantes de computadores para resolver essas questões. A IBM criou bibliotecas de sistemas operacionais e de aplicativos, evitando o retrabalho de desenvolvimento de um mesmo programa por usuários distintos.¹⁶ Todavia, embora os sistemas operacionais fossem reutilizados, a biblioteca de aplicativos tinha pouca função. Esses programas serviam mais como exemplos, sendo raramente reutilizados. O projeto do sistema era geralmente modificado e o *software* recodificado, atendendo às necessidades de um usuário em particular.

Com o modelo 360, a IBM unificou o sistema operacional (OS/360) de suas linhas de produtos, gerando uma significativa base instalada de computadores similares. Esse equipamento foi o primeiro a empregar o conceito de uma arquitetura de família, que consistia, originalmente, em seis computadores que podiam usar os mesmos *softwares* e periféricos. Seu lançamento serviu para consolidar o poder de mercado da empresa, através da manipulação da compatibilidade entre esses equipamentos a seu favor, gerando um processo de captura dos consumidores (efeito *lock in*).

Um programa de computador escrito para um modelo da série precisaria apenas ser recompilado e não reescrito para rodar numa outra máquina IBM. Dessa forma, no momento de trocar os equipamentos por máquinas de maior capacidade computacional, por exemplo, se optaria por um modelo da mesma família IBM, uma vez que os custos de mudança seriam menores em comparação com a opção de aquisição de um computador que não pertencesse à família IBM. Essa estratégia garantia fidelização e maior estabilidade de seus clientes.

A compatibilidade entre os modelos da linha permitiu a utilização de um mesmo pacote de *software* aplicativo por diversos usuários, embora, ao longo da década de 1960, uma parte representativa dos pacotes de *software* aplicativo ainda fosse ofertada pela IBM e outros fabricantes de *hardware*. Esses pacotes eram vendidos de forma casada com o equipamento, estando incluído no custo deste último.

A decisão da IBM em fornecer separadamente o *software* de seus equipamentos permitiu que as empresas desenvolvedoras de *software* entrassem nesse mercado, competindo em qualidade com o produto da IBM.¹⁷

O modelo de negócios das empresas independentes desenvolvedoras de *software* aproximava-se da indústria de bens de capital. Elas despendiam altos gastos em *marketing*, na criação de uma rede de suporte pré-vendas e pós-vendas e no fornecimento contínuo de manutenção do produto. A abordagem ao cliente era

¹⁶O legado deixado por essas bibliotecas foi a visão do software como um bem livre, sendo ofertado de graça.

¹⁷Não há um consenso quanto aos motivos geradores dessa decisão. Para alguns a empresa vislumbrava futuros problemas legais caso mantivesse essa estratégia – inquéritos contra a IBM baseados nas leis antitruste norte-americanas já haviam sido feitos. Outros argumentam que a empresa buscava concentrar suas atividades no hardware.

feita de forma direta, no que destacava-se também a IBM, com uma expressiva e experiente força de vendas.

Apesar das expectativas de decolagem da indústria de *software* ao longo da década de 1970, sua consolidação só ocorreu na década seguinte. Sua estrutura ainda era muito fragmentada, com um grande número de novos entrantes.

Nesse período, todavia, o mercado tomou forma, tendo sido definidos seus segmentos e atores. Uma infinidade de pacotes aplicativos surgiu nesse período, os quais podiam ser divididos em dois tipos: aqueles voltados para mercados verticais (por exemplo, bancos, telecomunicações etc.) e os de caráter geral (especializados em determinada função ou processo de negócio como contabilidade, folha de pagamentos, recursos humanos etc.).

Três tipos de empresas obtiveram êxito nesse setor: fabricantes de *hardware*, empresas independentes de *software* (ISV) e vendedores *turnkey*. Os fabricantes de *hardware*, que entraram nesse mercado como consequência quase que natural do fim da venda casada de *hardware* e *software*, viam esse negócio como uma fonte extra de receitas. Em contraste, as empresas independentes de *software* se constituíam em firmas novas, cuja atividade principal era o desenvolvimento de pacotes aplicativos, possuindo um longo caminho para consolidar sua reputação. Já os vendedores *turnkey* ofertavam a solução completa, integrando *software* e *hardware* e proporcionando suporte contínuo. Estes últimos cresceram em mercados verticais, como, por exemplo, varejo.

A Consolidação da Indústria de Software (1980/95)

A indústria de *software* passou por importantes transformações ao longo da década de 1980. Nesse período, o segmento de *software* pacote para o mercado corporativo começou a desenvolver características de uma indústria madura, com a estabilização de algumas empresas independentes de *software* nesse mercado.

Ao mesmo tempo, o surgimento dos microcomputadores abriu um novo mercado para as empresas de *software*. Já no início da década de 1990, posições monopolistas se apresentavam nos principais segmentos desse mercado, num processo de concentração atribuído aos fortes efeitos de rede prevaletentes no setor, consolidando padrões *de facto*.

Paralelamente, houve um renascimento de empresas prestadoras de serviços. Abriu-se espaço para uma nova modalidade de provedores de serviços de TI com a crescente dificuldade de administração da infra-estrutura de TI das grandes corporações, em função das recorrentes inovações tecnológicas.

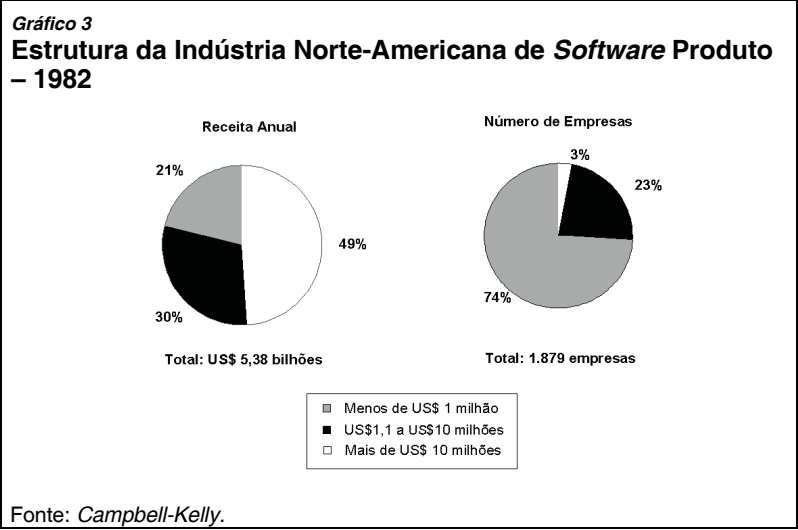
A partir do final da década de 1970, o setor de *software* pacote iniciou uma trajetória exponencial de expansão, atingindo taxas de crescimento anuais da ordem de 40%. Em meados da década de 1980 havia, catalogados, cerca de oito mil produtos de três mil vendedores distintos.

Nesse período ficou clara a supremacia dos Estados Unidos nessa indústria, respondendo por 2/3 do mercado mundial de *software*. Ao considerar o segmento de *software* pacote, sua participação representava 95% da oferta mundial. As empresas dos demais países limitavam-se ao desenvolvimento de *software* sob encomenda e serviços. A primazia das empresas norte-americanas é explicada pelas elevadas barreiras à entrada erguidas, ocupando mercados, ganhando escala e implantando padrões proprietários.

A estrutura da indústria norte-americana era piramidal.¹⁸ Em 1982, 3% das empresas eram de grande porte, respondendo por 49% do faturamento total do setor. As empresas médias representavam 23% do número total de firmas, com uma participação de 30% nas receitas totais. Já as empresas pequenas formavam a base da pirâmide – 74% das empresas se enquadravam nessa categoria –, mas sua participação no faturamento da indústria era pequena (21%).

Um importante acontecimento desse período foi o advento do microcomputador. Seu elemento crítico era o microprocessador (criado em 1971), que, quando combinado com *chips* de memória e periféricos, alcançava o desempenho de um computador. No final da década de 1970, já se encontravam microcomputadores no mercado, ofertados pela Apple, Commodore e Tandy.

A IBM lançou seu primeiro modelo em 1981, o IBM PC, que combinava um nível razoável de poder computacional e um sistema



¹⁸Essa estrutura é válida até hoje, prevalecendo elevadas taxas de nascimento e mortalidade entre as pequenas empresas. Tal fato é explicado em grande parte pela baixa necessidade de recursos para se criar uma empresa de software. Além disso, o ambiente dinâmico do setor abre a todo momento novos mercados, estimulando o surgimento de novos entrantes.

operacional que facilitava o desenvolvimento de aplicativos. Logo no primeiro ano foram vendidas 13 mil unidades. A novidade era sua arquitetura aberta, o que permitiu a criação de um mercado de “clones”. O rápido crescimento da base instalada de microcomputadores IBM e IBM-compatíveis criou um mercado homogêneo de sistemas operacionais e aplicativos para essas máquinas de tamanho sem precedentes na recente história dessa indústria. O IBM PC disseminou o poder de processamento nos departamentos das corporações.

Todavia, o modelo de negócios desses aplicativos se diferenciava daquele válido para produtos voltados para o mercado corporativo que rodavam em *mainframes* e minicomputadores. Os novos produtos deveriam ser intuitivos e com uma interface amigável, dado o baixo conhecimento em informática do seu usuário médio. Além disso, a instalação deveria ser rápida, não exigindo nenhum custo de implementação.

Adicionalmente, os preços e o volume de vendas eram significativamente diferentes. Os preços dos *softwares* para *mainframes* e minicomputadores giravam entre US\$ 5 mil e US\$ 200 mil, sendo vendidas algumas centenas de unidades. Enquanto isso, o *software* aplicativo para microcomputadores era vendido por algo entre US\$ 50 e US\$ 500, alcançando vendas de dezenas de milhares de unidades.

Dessa forma, também os canais de comercialização eram diferentes. A venda direta, utilizada no caso do aplicativo voltado para o mercado corporativo, era impraticável para o *software* para microcomputadores. Os primeiros pacotes de aplicativos para PC foram vendidos por meio de ordens de compra enviadas pelo correio. Com o crescimento do mercado, lojas de varejo especializadas começaram a surgir.

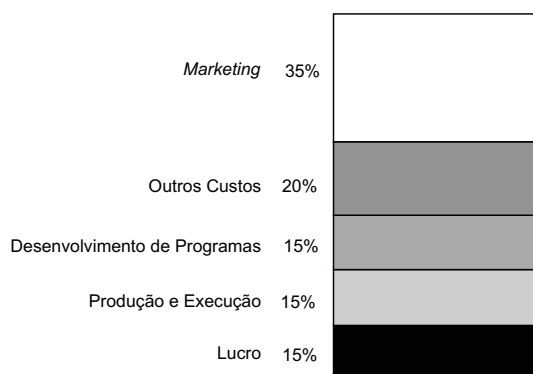
A maioria das empresas desenvolvedoras de *software* aplicativo para PCs que surgiram no período era formada por um ou dois programadores, os quais dispunham de capacidade técnica, mas não de habilidades em *marketing* e distribuição. Como resultado, duas figuras surgiram como intermediários entre essas empresas e o setor de varejo: o editor e o distribuidor (atacadista).

O elo crucial na cadeia de fornecedores de *software* era o editor. Seu papel tem sido subestimado, concentrando-se as atenções na criatividade do desenvolvedor. Cabia ao editor a tarefa, nem um pouco simples, de transformar o *software* em um produto aceito pelo mercado, o que envolvia altos investimentos em *marketing* e atividades promocionais, sendo o poder da marca e o esforço de divulgação muitas vezes determinantes nesse mercado. O Gráfico 4 apresenta a estrutura de custos típica de um negócio de *software* pacote.¹⁹

¹⁹Fred Gibbons, presidente da Software Publishing Corporation, dizia que as barreiras à entrada nesse mercado eram marketing, marketing e marketing [Campbell-Kelly (2003, p. 211)].

Gráfico 4

Estrutura de Custos do Modelo de Negócios de Software Pacote



Fonte: Campbell-Kelly.

Prevalencia uma certa simbiose entre o editor e a empresa desenvolvedora de *software*. Muitos produtos eram incorporados pelo primeiro. Por exemplo, o banco de dados Dbase II foi criado por um desenvolvedor independente (Wayne Ratcliffe) e publicado pela Ashton-Tate. Entre 1981 e 1983, as receitas dos editores de pacotes de *software* para PC subiram de US\$ 70 milhões para US\$ 486 milhões.

Esse mercado passou por um intenso processo de concentração. Logo no início da década de 1980 destacavam-se três empresas, cada qual com um produto específico líder no mercado: a Microsoft, que detinha o sistema operacional utilizado nos computadores pessoais da IBM e seus clones; a Lotus, com planilhas eletrônicas; e a Ashton Tate, cujo principal produto era o banco de dados Dbase. Em 1989, juntou-se a essas três a WordPerfect, que dispunha de um processador de texto.

Entre essas empresas, a Microsoft apresentou uma preeminente trajetória, liderando majoritariamente o mercado de sistemas operacionais e aplicativos para PCs já na década seguinte.²⁰ A decisão da IBM de disponibilizar o sistema operacional da Microsoft junto com sua máquina deu a esta última uma enorme vantagem competitiva, pois o domínio desse segmento permitiu-lhe a conquista de outros nos anos seguintes. Isso porque o sistema operacional ocupa uma posição primordial na arquitetura da máquina, sendo que o desenvolvimento de qualquer aplicativo se dá com base nele. A Microsoft se aproveitava do acesso privilegiado às informações sobre suas características técnicas para lançar antecipadamente seus aplicativos, os quais já incorporavam características apropriadas à nova versão do sistema operacional.

Aliado a isso, a Microsoft adotou uma estratégia que envolvia anúncios antecipados de lançamentos de novas versões de seus

²⁰Uma detalhada descrição da estratégia e dos métodos adotados pela Microsoft até assumir a posição monopolista que ostenta hoje no segmento de software para PCs é apresentada em Roselino (1998).

produtos, impactando as vendas de seus concorrentes. O simples anúncio de lançamento de um novo produto já provocava efeitos como a aceleração da depreciação de produtos concorrentes e a intensificação das barreiras à entrada de novos produtos. Estes últimos devem superar não apenas aqueles que estão no mercado, como também os já anunciados, mas ainda indisponíveis.

Em 1993, a Software Publishers Association informou que os aplicativos do MS-Windows superaram pela primeira vez em vendas os programas para o MS-DOS.

Junto com o PC, foi lançada, pela Apollo, a primeira estação de trabalho, seguida, em 1982, pelo modelo da Sun. A máquina, que era um híbrido entre o computador pessoal e o minicomputador, beneficiou-se, como o PC, dos avanços dos microprocessadores, tanto tecnicamente (com tamanho cada vez menor e melhor desempenho) quanto economicamente (custos decrescentes). Ao mesmo tempo, como no caso do minicomputador, os periféricos agregados à central de processamento da estação de trabalho proporcionavam um alto desempenho computacional e saídas gráficas, inclusive no monitor.

Enquanto a IBM optou por comprometer a capacidade computacional com o PC para entrar no mercado de *desktops*, os fabricantes de estação de trabalho atraíam usuários mais sofisticados tecnicamente (por exemplo, engenheiros), que, caso contrário, usariam minicomputadores e *mainframes*.

A empresa fabricante de estação de trabalho de maior sucesso, a Sun, optou pela adoção do Unix (sistema operacional desenvolvido pela Bell Laboratories). Além disso, a sua estratégia, similar à da IBM, de manter o mesmo sistema operacional nas famílias de computadores, permitindo que os aplicativos fossem portáteis para gerações futuras, estimulou o desenvolvimento de empresas independentes de *software* para essas máquinas.

Por fim, a incorporação de novas tecnologias surgidas no mercado à infra-estrutura de TI de uma empresa era um processo quase sempre caótico. A crescente variedade e volume de *hardware* e *software* alavancou os problemas de compatibilidade e a complexidade em administrar a base instalada. Muitas vezes, a empresa não dispunha de recursos humanos qualificados para essa tarefa e/ou capazes de acompanhar as inovações ocorridas no mercado.

As crescentes dificuldades estimularam o florescimento de provedores de serviços profissionais de TI. Alguns ofereciam serviços que iam desde a construção do departamento de processamento de dados e gerenciamento das especificações estipuladas pelo cliente até a subcontratação para desenvolvimento de *software* e, eventualmente, do *staff* das operações, embora a maioria prestasse serviços menos amplos.

Previamente à análise do mercado de *software*, cabe apresentar algumas forças intrínsecas a esse setor, que são determinantes para a sua dinâmica.

Mercado

Observa-se na indústria de *software* segmentos fortemente concentrados, onde poucas empresas detêm uma elevada participação de mercado. O exemplo clássico é o sistema operacional Windows, da Microsoft.

Barreiras à Entrada: Especificidades do *Software*

A estrutura oligopolista é explicada, em grande parte, pelas barreiras à entrada erguidas em função das economias de escala e de rede presentes nessa indústria, em distintos graus nos diversos segmentos.

O *software* pacote apresenta um custo de reprodução próximo de zero. Sua “produção” não envolve um processo fabril, sendo nada mais do que a mera replicação de linhas de código previamente desenvolvidas. A predominância de custos fixos torna a escala um fator crítico para a sustentabilidade de uma empresa desenvolvedora de produto. Uma vez amortizados os custos de desenvolvimento, os ganhos são elevados. Os ganhos extraordinários obtidos, por sua vez, são totalmente reinvestidos em P&D para melhoria e aperfeiçoamento do produto (atualizações e novas versões), permitindo contínua manutenção de sua competitividade.

Economias de Escala e de Rede

Como a escala é um fator crítico nesse mercado, o sucesso de uma empresa de produto está relacionado à sua habilidade em abstrair o conhecimento obtido durante o processo de desenvolvimento do *software*. Elementos específicos devem ser sacrificados em função de maior aplicabilidade a situações diversas, aumentando sua potencial base de clientes.

A facilidade de replicação de um produto de *software* torna a proteção da propriedade intelectual um importante instrumento para a indústria. A falta de uma proteção efetiva dificulta a manutenção da competitividade de uma empresa, pois o uso de cópias não autorizadas do produto desestimula o seu crescimento. No Brasil, ele é protegido de forma semelhante ao direito autoral pela Lei 9.609, de 19 de fevereiro de 1998.

Além do custo marginal ínfimo, prevalecem nesse setor externalidades advindas da arquitetura de rede. O princípio básico do efeito de rede é anunciado pela Lei de Metcalfe, segundo a qual

o valor da rede eleva-se com o quadrado do número de usuários. O telefone e a internet são exemplos típicos desse princípio.

Quando alguém se integra à rede, ela fica melhor e maior, beneficiando a si próprio e aos demais integrantes. Esse efeito é auto-alimentado na medida em que a expansão da base de usuários faz com que um número maior de pessoas acredite valer a pena adotar o sistema, gerando um ciclo virtuoso de crescimento.

A arquitetura de rede gera fortes economias de escala: a) o valor do produto para cada usuário aumenta à medida que a rede aumenta; b) uma vez que participantes de uma rede decidem adotar um novo produto, a motivação de outros em também adotá-lo se torna maior; e c) permite a rápida difusão de idéias e notícias.²¹

O conceito de efeito de rede pode ser estendido à indústria. Em algumas indústrias consolida-se uma estrutura de rede, prevalecendo fortes interligações entre os diversos atores (fornecedores, fabricantes, distribuidores, consumidores etc.).

A economia de rede ergue enormes barreiras à entrada, consolidando padrões *de facto*. A hostilidade de um mercado dominado por um padrão proprietário à inovação torna-se maior quando seus atores estão interligados. Cada participante irá mudar para um novo produto apenas quando acreditar que os outros também o farão.

A existência de economia de rede desloca o foco das características e funções do produto para a rede em que ele está inserido. O sucesso da inovação depende de sua habilidade em conseguir a adesão de um número de participantes suficiente para dar suporte ao seu produto. A escolha de cada ator variará em função do papel desempenhado nessa rede (por exemplo, os competidores, parceiros, reguladores, varejistas ou consumidores). A empresa deve descobrir como fazer para que os diferentes participantes mudem seus comportamentos e adotem sua inovação.

No setor de *software*, prevalece a economia de rede. Quanto maior o número de usuários de um determinado programa de computador, maiores serão os benefícios gerados. Por exemplo, uma empresa desejará que todos os seus funcionários utilizem o mesmo aplicativo em seus computadores, permitindo a troca e compartilhamento de informações e conhecimentos (interoperabilidade). Isso transborda para o nível extra-empresa, em que o padrão adotado por determinada empresa tende a ser o mesmo daquele adotado por empresas com as quais tenha ou possa vir a ter relações (fornecedores, clientes, possíveis empresas a serem incorporadas etc.). Além disso, a utilização maciça de um determinado *software* cria uma disponibilidade de recursos tais como manuais, mão-de-obra treinada, serviços de suporte, treinamento etc.

²¹ Por outro lado, a configuração de rede apresenta um caráter segregador, excluindo aqueles que não estão inseridos nessa estrutura.

A troca de um produto de *software*, mesmo que por outro mais eficiente, pode envolver altos custos de mudança, os quais são tanto maiores quanto maior for a rede de valor criada em torno do produto.

As empresas de *software* geralmente despendem um grande esforço desenvolvendo produtos que tenham uma complementaridade, dêem suporte ou otimizem a funcionalidade dos produtos já existentes. Dessa forma, os vendedores do produto conseguem ganhar um mercado maior, o que, por outro lado, torna mais custoso para o consumidor trocar seu produto por outro. Por exemplo, empresas desenvolvedoras de aplicativos adotarão a plataforma dominante, visando a um mercado consumidor potencialmente maior, reforçando o padrão *de facto*.

Todavia, deve-se ressaltar que a magnitude dessas barreiras impostas pelo baixo custo marginal e a economia de rede variam entre os diversos segmentos desse setor. De modo geral, quanto maior o custo de desenvolvimento, o tamanho do mercado e menos customizado for o *software*, maiores serão as barreiras à entrada.

Dessa forma, considerando as barreiras apresentadas acima, podem ser identificadas duas estratégias distintas de entrada nessa indústria: a) através de inovações complementares aos produtos existentes, reforçando o padrão e se beneficiando de toda a rede de valor envolvida; e b) através de uma inovação descontínua, em que se propõe uma mudança de padrão com uma tecnologia inovadora tirando proveito de um mercado ainda em construção.

O desenvolvimento de *software* é normalmente realizado em quatro etapas, subseqüentes à fase de análise da situação-problema que se pretende tratar, quais sejam: especificação de requisitos; projeto e implementação; teste e validação; e manutenção. A ordem seguida é sempre essa, mas existem variações quanto ao número de vezes que essa seqüência é percorrida. Uma única vez, como nos desenvolvimentos tradicionais, ou em interações sucessivas, que buscam apresentar uma primeira solução mais simples, que vai sendo refinada progressivamente. Esse é um dos fatores que distingue as metodologias de desenvolvimento.

Tais metodologias têm sido criadas para, por um lado, suprir deficiências na especificação do produto desejado, a qual comumente muda várias vezes durante o desenvolvimento, e, por outro, propiciar maior controle sobre o processo de desenvolvimento e sua aderência às especificações.

Certificação

O tamanho e a complexidade crescentes dos produtos de *software* têm trazido a necessidade premente de serem criadas e aperfeiçoadas técnicas de controle sobre o processo. Tem-se verificado que apenas uma pequena parcela dos desenvolvimentos é concluída dentro do prazo e do custo estimados, sendo as diferenças em relação às estimativas originais extremamente significativas. De forma análoga, o atendimento às especificações também deixa muito a desejar, sendo muito elevado o índice de desenvolvimentos cancelados, assim como o número de produtos que são apenas parcialmente utilizados.

O custo do desenvolvimento do *software* – associado aos recursos disponibilizados para a atividade – e o tempo estimado para a sua realização são de natureza quantitativa. Já as técnicas de engenharia de *software*, as metodologias de desenvolvimento e as técnicas de programação e teste são, em sua maioria, qualitativas. Esse descompasso traz dificuldades à atividade de gerenciamento do desenvolvimento de *software*. Buscando solucionar esse problema, alguns processos de medição de *software* têm sido criados. Nesse sentido, são pioneiras na implantação de métodos de avaliação de *software* organizações de defesa dos Estados Unidos, da Inglaterra e da Alemanha, assim como empresas de telecomunicações desses países.

Um grande número de padrões de avaliação de qualidade de *software* tem sido criado, sendo os mais conhecidos o *capability maturity model* (CMM) e o ISO 15504. Ambos colocam em foco o grau de maturidade exibido por uma dada empresa de *software* em suas habilidades de gerenciar o processo de desenvolvimento e de manutenção em toda a organização.

O CMM foi desenvolvido pelo Software Engineering Institute (SEI) centro de P&D na Universidade Carnegie Mellon patrocinado pelo Departamento de Defesa dos Estados Unidos, cujo principal objetivo é colaborar para o aprimoramento das habilidades de terceiros em engenharia de *software*. Seu meta pode ser traduzida como “o *software* certo, fornecido sem defeitos, no prazo e dentro do custo, sempre” (*the right software, delivered defect free, on time and on cost, every time*). Sua certificação e a prévia consultoria têm sido largamente utilizadas como forma de garantir a excelência do processo de desenvolvimento de *software* das empresas.

O padrão ISO 15504, também conhecido como Spice (*software process and capability determination*), fornece especificações e modelo para determinação da maturidade das empresas desenvolvedoras de *software*. É uma iniciativa de origem européia similar ao padrão *software capability maturity model* (SW-CMM), aplicada ao desenvolvimento de *software*.

Como tentativa de harmonizar a proliferação de normas e padrões CMM – para desenvolvimento, aquisição etc. –, foi criado o

capability maturity model integration (CMMI), cuja versão corrente foi lançada em 2002. Seus modelos contemplam as melhores práticas de engenharia de *software*, engenharia de sistemas e desenvolvimento integrado de produto e processo, buscando compatibilidade com a norma Spice. O grupo responsável pela condução do projeto CMMI é constituído por representantes do SEI, da indústria e do governo norte-americano.

O CMMI apresenta duas estruturas de classificação – uma em níveis crescentes, tal qual o SW-CMM precursor, e uma contínua, da mesma forma que o Spice. A classificação em níveis do CMMI é um pouco diferente da classificação SW-CMM, bastante difundida. A nova classificação e seus focos são apresentados de forma resumida a seguir:

NÍVEL DE MATURIDADE	FOCO
5 – Otimizado	Melhoria contínua do processo
4 – Gerenciado quantitativamente	Gerenciamento quantitativo
3 – Definido	Padronização do processo
2 – Gerenciado	Gerenciamento básico de projeto
1 – Inicial	

Já o CMMI contínuo possui categorias de processo semelhantes à norma ISO/IEC 12207 (*standard for information technology – software life cycle processes*), relativas a: a) gerenciamento do processo; b) gerenciamento do projeto; c) engenharia; e d) suporte.

De forma semelhante à Spice, o CMMI contínuo possui seis níveis, que vão do zero (imaturo) ao cinco (plenamente maduro), havendo uma correspondência determinada entre a categorização em estágios e a melhoria dos processos de uma organização em cada uma das áreas-chave do padrão. Esses níveis são apresentados resumidamente a seguir:

NÍVEL DE CAPACIDADE	CCMI CONTÍNUO
5	Em otimização
4	Quantitativamente gerenciado
3	Definido
2	Gerenciado
1	Executado
0	Incompleto

A obtenção de certificações de reconhecimento internacional como a CMM e a Spice tem sido apresentada pelas empresas do setor como garantia de qualidade dos seus processos, na medida em que essas certificações provam o grau de controle praticado. Isso

assegura aos clientes, de acordo com o nível alcançado, a aderência dos produtos gerados às especificações, a capacidade de repetição do processo de desenvolvimento sem queda de qualidade e o contínuo aperfeiçoamento da organização, entre outros.

A garantia prestada pela certificação é especialmente importante no caso dos serviços de *outsourcing*, uma vez que, em geral, não há um produto a ser testado *a priori* antes da contratação. Entretanto, seu impacto sobre as empresas de produto também se verifica.

A utilização da certificação como barreira à entrada é sensível em mercados desenvolvidos, limitando o acesso de pequenas empresas, excluídas pelos altos custos envolvidos nos processos de consultoria e de transformação de processos prévios à certificação. É também uma barreira a ser transposta por países que almejam lançar-se ao mercado externo.

Mercado Mundial

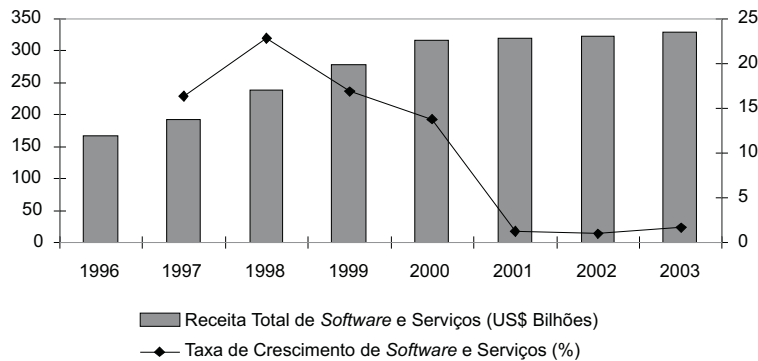
A indústria de *software*, como qualquer setor tipicamente de alta tecnologia, está em constante mudança. Seu caráter dinâmico torna sua esquematização uma tarefa desafiadora. As intermitentes inovações ocorridas tanto intra-indústria quanto em outros setores de tecnologia (por exemplo, microeletrônica e telecomunicações) geram a todo momento novos mercados e oportunidades para empresas do setor.

Essa forte dinâmica cria um ambiente propício ao surgimento de novas empresas. Isso, associado à baixa necessidade de capital, explica o grande número de empresas de pequeno e médio portes prevalecente nessa indústria em todo o mundo. Paradoxalmente, verificam-se posições monopolistas e segmentos altamente concentrados, em função das elevadas barreiras à entrada erguidas em mercados já maduros. Além disso, verifica-se a tendência de absorção, por aquisição, de novas empresas pelas grandes já consolidadas no mercado. Um exemplo foi a compra da Rational – inovadora em metodologia e ferramentas de desenvolvimento – pela IBM.

O mercado dos Estados Unidos destaca-se pela sua magnitude, pois as maiores empresas do setor estão localizadas no país. Segundo estudo realizado pelo U.S. Department of Commerce (2003), o mercado de *software* e de serviços somava US\$ 328,8 bilhões em 2003, representando um crescimento de 1,6% em relação ao ano anterior. As taxas de crescimento anuais negativas observadas nos anos anteriores são atribuídas ao baixo crescimento da economia no período e ao “estouro” da bolha “ponto com”²² (ver Gráfico 5).

²²Queda brusca do valor de mercado das empresas ligadas à internet, então sobrevalorizadas.

Gráfico 5
Evolução das Receitas de Software e Serviços do Mercado Norte-Americano – 1996/2003



Fonte: U.S. Department of Commerce (2003).

As estatísticas sobre o setor de *software* são escassas e pouco confiáveis, pois o seu caráter dinâmico impede o estabelecimento de uma classificação homogênea, dificultando a análise dos dados disponíveis.

Produtos de Software

Segundo a revista *Software Magazine*, as receitas das 500 maiores empresas de *software* em 2002 somavam US\$ 289,7 bilhões, representando uma retração de 18% em relação ao ano anterior. A baixa *performance* desse mercado nos últimos anos é atribuída em parte ao fraco desempenho da economia mundial, levando as empresas a adiar investimentos.

A baixa *performance* das vendas de licenças indica um esgotamento do modelo até então praticado pelas empresas independentes desenvolvedoras de *software* de licenças perpétuas. Novos meios de comercialização dos produtos têm surgido, e as empresas vêm adotando cada vez mais outros tipos de contratos, como, por exemplo, a subscrição. O preço do produto é diluído ao longo do tempo, e a empresa passa a receber mensalmente um valor pela utilização de seu produto, permitindo maior estabilidade de seu fluxo financeiro, enquanto para o usuário fica mais fácil o endereçamento do valor do *software*, pago na medida da efetiva utilização.

Nesse novo modelo, a distribuição do *software* para os usuários é feita por uma rede, ao invés de se exigir que eles o rodem em seu próprio *hardware*. Esse é o conceito de *application service provider* (ASP), em que o aplicativo é hospedado em um servidor em nome do cliente. Por exemplo, as receitas do negócio de hospedagem da Oracle cresceram mais de 80% no último trimestre de 2003.

A seguir serão analisados separadamente cada um dos segmentos desse mercado: infra-estrutura, ferramentas e aplicativos.

Segmento de Infra-Estrutura

Os produtos ligados mais diretamente à parte física da infra-estrutura de TI (sistemas operacionais, gerenciadores de rede etc.) passam quase que despercebidos para muitos. Em alguns casos, são vendidos de forma casada com os equipamentos. Entretanto, desempenham uma função estratégica, sendo essenciais tanto para o funcionamento de qualquer equipamento quanto para o desenvolvimento de aplicativos. Prevalece nesse mercado uma estrutura altamente concentrada em algumas grandes firmas globais – BEA, Computer Associates, HP, IBM, Microsoft, Novell, Oracle, Sun, Unisys, entre outras.

O mercado mundial de sistemas e subsistemas operacionais somava US\$ 18,6 bilhões em 2002, segundo a IDC, destacando-se o grande poder de mercado da Microsoft, cuja participação na venda de novas licenças em sistemas operacionais para servidores era de 55% e de 93,8% para computadores pessoais em 2002 (ver Tabela 2).

Ganha expressão, nesse contexto, o movimento de *software* livre, apontado como forte ameaça à liderança da Microsoft, pois já se observa, de fato, um representativo índice de penetração do Linux em servidores. Uma análise mais detalhada do conceito de *software* livre, bem como das suas implicações no setor, é feita mais à frente neste trabalho.

Um mercado que apresenta elevadas taxas de crescimento é o de *software* de segurança. Segundo a *Business Week* (21.06.04, p. 84), o total desse segmento, incluindo produtos e serviços, era de

Tabela 2
Participação nas Vendas de Novas Licenças de Sistemas Operacionais – 2002
(Em %)

	SERVIDORES	COMPUTADORES PESSOAIS
Microsoft	55,0	93,8
Linux	23,0	2,8
Unix	11,0	0,0
Netware	9,9	0,0
MacOS	0,0	2,9
Outros	1,1	0,5

Fonte: IDC.

US\$ 27 bilhões, apresentando uma taxa de crescimento anual de 19%. Entre as principais empresas de *software* de segurança figuram a Symantec, a NetScreen e a SuftControl.

As volumosas perdas amargadas por empresas que tiveram suas redes invadidas por vírus têm tornado crescente a preocupação com segurança. Segundo pesquisa realizada pelo Gartner Group com CIO (*chief information officer*) de grandes empresas sobre as prioridades dos gastos em TI, o *software* de segurança foi apontado como o primeiro nos dois últimos anos consecutivos.

Novos produtos de segurança têm surgido no mercado. São *softwares* e serviços que fornecem avisos de alerta antecipadamente. Ao contrário dos tradicionais *softwares* antivírus ou detectores de invasores, que criam barreiras ao redor da rede corporativa, esses novos sistemas percorrem a internet à procura de novos vírus, alertando com antecedência a empresa para a sua presença. Em lugar de esperar a entrada do vírus, o sistema constrói previamente defesas contra o que for detectado.

Segmento de Ferramentas

Esse segmento, assim como o de infra-estrutura, é pouco familiar para pessoas fora da área de TI. Os principais usuários de ferramentas são os próprios programadores de departamentos de processamento de dados das empresas ou empresas desenvolvedoras de *software*. Merecem citação também as fábricas de *software*, usuárias intensivas desse tipo de ferramentas.

A oferta nesse segmento também é bastante concentrada, sendo compartilhada por grandes *players* globais – Borland, Computer Associates, IBM, Intersystems, Microsoft, Progress, Sybase, Seagate Software, Oracle etc.

Os serviços Web podem ser definidos como um conjunto de padrões que possibilitam interações diretas entre aplicações através do uso de mensagens no formato XML em protocolos de internet. A maioria dos produtores de *software* vem suportando os padrões Web nos seus produtos atuais. Nos próximos anos esses padrões deverão ser evoluídos, sendo incorporados às novas versões dos produtos em desenvolvimento. Dessa forma, os padrões de serviços Web estarão presentes em múltiplos *softwares*, transformando a rede em um imenso computador distribuído.

Essa rede é composta de plataformas sobre a qual as aplicações são desenvolvidas. Duas plataformas concorrentes disputam a liderança desse mercado – Java e .Net –, cujo êxito depende do número de empresas e programadores que irão aderir a cada uma delas.

O Java, criado pela Sun, é um padrão aberto, que foi inicialmente adotado pela grande maioria de servidores de Web. Sua grande vantagem é propiciar que aplicativos desenvolvidos na linguagem Java possam ser executados em qualquer plataforma que disponibilize a máquina virtual Java. Por ser um padrão aberto, o usuário não fica preso a um único fornecedor, fato que motiva a competição e, conseqüentemente, reduz preços. Todavia, empresas que adotam o padrão Java, como IBM e Sun, tendem a acrescentar acessórios às versões das plataformas que vendem para justificar ágios sobre seus produtos. Mais, essas diferenças implicam que o aplicativo Java que roda em um sistema não possa ser executado em outro, o que reduz a atratividade dessa plataforma.

Já o .Net foi criado pela Microsoft como alternativa ao Java. Usa os mesmos padrões de serviços Web e possui também uma máquina virtual, a qual se integra perfeitamente ao Windows, consoante a estratégia adotada pela Microsoft de tirar vantagem de sua posição dominante na indústria de computadores pessoais. A empresa anunciou o lançamento no próximo ano de um poderoso conjunto de ferramentas visando facilitar o desenvolvimento baseado na plataforma.

Segmento de Aplicativos

Os aplicativos voltados para o mercado corporativo seguem uma lógica diferente daqueles orientados para o usuário doméstico de microcomputadores, sendo aqui analisados apenas os primeiros.

Os aplicativos corporativos geralmente surgem de demandas da indústria. Muitas vezes, o desenvolvimento de um *software* sob encomenda para determinada empresa transforma-se num produto, após certa abstração, sendo ofertado para o mercado como um todo.

A escolha da empresa demandante em favor do pacote ou do *software* sob encomenda dependerá da relação custo/benefício entre um *software* de caráter geral *versus* aquele que se ajuste perfeitamente às suas idiosincrasias.

O modelo de negócios desses aplicativos muito se aproxima da indústria de bens de capital. A venda do pacote vem acompanhada de uma parcela significativa de serviços associados, como customização, implementação, manutenção e treinamento, exigindo uma extensa rede de suporte pré-vendas e pós-vendas. Similarmente ao que acontece na indústria de bens de capital, seu crescimento é fortemente puxado pela demanda. Períodos de baixo crescimento econômico, no qual as empresas reduzem seus investimentos, afetam fortemente as vendas desses produtos.

Nesse segmento figuram duas grandes classes de pacotes: os de uso geral (interindústria), englobando produtos que administram uma função ou processo de negócio como contabilidade, folha de pagamentos, recursos humanos, relacionamento com fornecedores etc., cujos exemplos mais notórios são ERP, CRM e SCM; e os verticais (intra-indústria), voltados, por exemplo, para o setor financeiro, as telecomunicações etc., que demandam um profundo conhecimento do negócio do cliente. Entre as principais empresas atuantes no segmento de aplicativos destacam-se SAP, Oracle, J.D. Edwards, PeopleSoft, Microsoft e IBM, entre outras.

Muitas vezes, questões como arcabouço legal e estrutura tributária do país criam reservas de mercado naturais nesse segmento. Dessa forma, prevalece uma estrutura de mercado bastante fragmentada, com um representativo número de empresas nacionais que, todavia, parecem estar presas aos mercados de origem, sendo sua internacionalização um salto difícil de ser alcançado.

A década de 1990 foi extremamente fértil para empresas provedoras de serviços de TI. O *boom* tecnológico ocorrido no período foi acompanhado de um fabuloso crescimento dos gastos corporativos em TI mundialmente. Segundo dados da McKinsey, esse mercado passou de US\$ 175 bilhões no início da década para US\$ 525 bilhões em 2000.

Serviços de Software

Nesse período, consolidou-se uma nova forma de compra de serviços: o *outsourcing*, mercado que tem apresentado recorrentemente elevadas taxas de crescimento. Ao mesmo tempo, a busca pela redução de custos vem consolidando o modelo *offshore* (em outro país).

As maiores empresas que atuam nesse segmento são de origens distintas. Entre elas, encontram-se empresas tradicionalmente do setor de serviços (por exemplo, a Accenture), empresas advindas do departamento de TI de grandes corporações (por exemplo, a Atos Origin, que surgiu da Philips) e empresas fabricantes de *hardware* (que têm aumentado a participação das receitas de serviços em seu faturamento total). O interesse de fabricantes de *hardware* nesse setor de serviços pode ser explicado pela tentativa de preservar e/ou aumentar sua participação de mercado, além de elevar suas receitas. Ao celebrar um contrato de *outsourcing*, a empresa provedora do serviço absorve os equipamentos do cliente, mesmo que a tecnologia empregada seja de um concorrente. Quando há necessidade de atualização, porém, fica numa posição privilegiada para substituí-los por produtos próprios.

Segundo o Gartner Group, em 2003 o mercado mundial de serviços de TI atingiu a cifra de US\$ 569 bilhões, representando um

crescimento de 6,2% em relação ao ano anterior. A IBM destaca-se como a maior empresa do setor, com um faturamento de US\$ 42,6 bilhões, sendo que suas receitas de serviços ultrapassaram mais da metade do seu faturamento total em 2003, como mostra a Tabela 3.

Muitos provedores de serviços prosperaram, ao longo da década passada, implementando grandes pacotes de aplicativos corporativos (e.g., ERP, SCM e CRM). Além da instalação, proviam orientações sobre como implementar as mudanças nos processos de negócio para capturar as eficiências do novo *software* e desenvolviam aplicativos customizados, estreitando os laços com o cliente. O valor desses projetos normalmente girava em torno de US\$ 10 milhões, sendo baseados nos custos estimados da mão-de-obra e margens de lucro padrão, sem nenhuma forma de garantia dos benefícios gerados ao cliente.

Paralelamente, um crescente número de empresas alterou seu método de compra de serviços de TI, adotando o modelo de *outsourcing*.²³ A decisão a favor do *outsourcing* estava muito relacionada aos seguintes fatores:

- terceirização de atividades não críticas para a organização, a qual se concentrava em questões mais estratégicas e no seu *core business*;
- obtenção de melhores práticas de processo; e
- redução de custos.

De modo geral, os contratos de *outsourcing* apresentavam foco amplo e prazos longos, cobrindo operações de *data centers*, gerenciamento de estações de usuários (*desktops*) suporte (*help desk*), gerenciamento de redes e, mais recentemente, serviços como hospedagem e gerenciamento de aplicativos.

A constante pressão em busca de custos cada vez menores, associada ao desenvolvimento das telecomunicações, que reduziu as limitações geográficas entre os mercados, levou ao crescimento do modelo *offshore*. A escolha pelo provedor de serviços tomou uma escala global. Dessa forma, serviços de TI passaram a ser realizados em países como Índia e China, usufruindo do excesso de mão-de-obra qualificada e do menor custo presente nesses países.

²³A Kodak foi pioneira nesse processo. Em 1990, a empresa assinou um contrato de US\$ 250 milhões com a IBM, a Digital Equipment Corp. e a Businessland por um período de 10 anos, terceirizando toda a sua área de TI.

Em 2003, o faturamento das empresas indianas representava 1,4% do total do setor. Embora pouco significativo, suas receitas têm crescido regularmente – enquanto o faturamento total das empresas norte-americanas apresentou uma expansão de 4% entre 2002 e 2003, o das indianas elevou-se em 29% no mesmo período.

Tabela 3

Os 10 Maiores Provedores Mundiais de Serviços de TI por Faturamento – 2003

(Em US\$ Milhões)

EMPRESA	FATURAMENTO	PARCELA DO MERCADO (%)	FATURAMENTO EM 2002	PARCELA DO MERCADO EM 2002 (%)	VARIAÇÃO DO FATURAMENTO (%)
IBM	42.635	7,5	40.139	7,5	6,2
EDS	21.102	3,7	20.979	3,9	0,6
Fujitsu	16.123	2,8	14.483	2,7	11,3
Computer Sciences (CSC)	12.679	2,2	12.123	2,2	4,6
Hewlett-Packard	12.546	2,2	12.211	2,3	2,7
Accenture	12.150	2,1	11.514	2,1	5,5
Lockheed Martin	7.568	1,3	7.143	1,3	5,9
Northrop Grumman	7.392	1,3	6.700	1,2	10,3
Automatic Data Processing	7.220	1,3	7.043	1,3	2,5
Hitachi	7.134	1,3	6.408	1,2	11,3
Outras	422.454	74,2	396.917	74,1	6,4
Mercado Total	569.003	100,0	535.660	100,0	6,2

Fonte: Gartner Dataquest.

As empresas indianas são extremamente baseadas no modelo *offshore*, sendo 92% de suas receitas provenientes de clientes estrangeiros, relativas, em sua maior parte, a serviços de baixo valor agregado, como atividades de codificação (fábricas de *software*), testes e serviços de *call-center*. Algumas delas já apresentam porte representativo, destacando-se a Tata, a Wipro e a Infosys.

Os serviços de *outsourcing* têm-se tornado cada vez mais sofisticados. Atividades estratégicas das empresas passaram a ser terceirizadas, e a redução de custos *per se* deixou de ser o principal objetivo buscado num processo de *outsourcing*. Outras variáveis, tais como acesso a tecnologias no estado da arte, padronização de processos, *expertise* na área de conhecimento etc., entraram no rol dos motivos que levam as empresas a buscar o *outsourcing*.

Porém, quanto maior a complexidade e mais estratégica for a atividade a ser terceirizada, mais importante se torna a marca do provedor de serviços. Destacam-se aqui os contratos de BPO realizados entre grandes corporações e os principais provedores de serviços como IBM e Accenture.

Os contratos dessa nova modalidade de *outsourcing* são mais extensos e o período de negociação é mais longo, demandando das empresas do setor experiência e recursos para cobrir os enormes gastos em vendas. Algumas vezes, o período de negociação de um projeto de BPO pode durar entre seis e 12 meses, levando a uma mobilização de pessoas sem a garantia de que o negócio será fechado.

A forte competição entre os provedores de serviços e o alto poder de barganha dos clientes têm imposto fortes pressões sobre as margens das empresas do setor. Antes de terceirizar, muitos clientes já fizeram todos os ajustes e padronizações internamente, estando num baixo nível de custo, o que leva a pressionar os preços das empresas de serviços.

Além disso, os clientes têm acumulado experiência na negociação de contratos, sendo crescentes as exigências feitas. Há quatro anos 85% dos contratos de consultoria eram cobrados por hora, número que hoje caiu para 20%. Atualmente a maioria dos contratos de prestação de serviços é atrelada aos resultados sobre o ganho de *performance* do cliente obtido com a terceirização (eficiência, redução de custos ou crescimento das vendas).

Diante desse quadro, cabe às empresas do setor o desafio de elevar sua produtividade como forma de ganhar mercado e elevar suas margens. Embora o mercado *offshore* continue sendo uma opção para obter mão-de-obra barata, isso não parece ser suficiente. Vêm, então, sendo adotadas iniciativas como a criação de bibliotecas de componentes (módulos de *software* reutilizáveis), diminuindo o esforço humano e o tempo necessário para o desenvolvimento do programa encomendado. Algumas empresas desenvolveram soluções padronizadas, voltadas para mercados verticais (setor bancário, telecomunicações, automotivo etc.) e/ou atividades específicas (contabilidade, recursos humanos etc.), diminuindo o prazo e o número de funcionários necessários à implementação do serviço. Outras alcançaram maior produtividade com a automatização de tarefas antes realizadas por um grande número de programadores.

Apesar de a demanda por serviços de TI ser crescente, as barreiras de acesso a esse mercado têm se tornado cada vez maiores. Fatores como casos de sucesso no currículo das empresas de TI (*track record*), certificação (principalmente CMM) e marca são condições *sine qua non* para fechar grandes contratos. Já o espaço de atuação de empresas de menor porte, geralmente, está limitado a subcontratações realizadas por grandes empresas do setor.

O crescente processo de globalização da economia tem impactado a forma como os serviços de TI são ofertados e adquiridos, tomando uma escala global. Esse novo modelo – denominado *global delivery model* – envolve uma proposição de maior valor agregado e mais ampla que o conceito de *offshore*. As empresas podem utilizar recursos de TI de qualquer país do mundo, ao mesmo tempo que a competitividade dos provedores de serviço passa a ser baseada na capacidade de ofertar aos clientes uma alocação ótima de recursos numa esfera global.

Computação como *Utility*

Recentemente, um novo conceito de serviço tem sido ofertado pelas maiores empresas do setor: computação sob demanda (*on demand*), que envolve a venda de poder computacional (de processamento) como *utility*, isto é, implementação da computação como serviço público (por exemplo, eletricidade e água).

A computação sob demanda irá permitir que as organizações excluam de sua estrutura de capital os altos investimentos em TI. Segundo matéria publicada pela *Informática Hoje*, em junho de 2004, a computação vista como *utility* transforma-se em excelente oportunidade para:

- oferecer TI como um serviço, não como uma caixa ou um bem físico;
- prover TI como um bem adequado, pelo preço justo e quando for necessário;
- responder rapidamente às mudanças e suportar as necessidades do negócio; e
- contabilizar os custos de TI de forma proporcional à qualidade do serviço prestado.

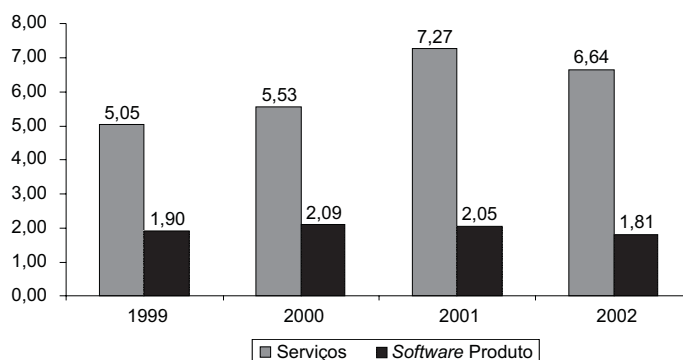
Entretanto, ainda há um longo caminho a ser percorrido para a consolidação desse serviço, já que pouco avanço foi feito acerca da visão sob demanda.

O setor de *software* vem apresentando historicamente elevadas taxas de crescimento quando comparadas com a *performance* da economia. Segundo a *Série Estudos Tecnologia da Informação*, o mercado nacional de *software* atingiu a cifra de US\$ 8,45 bilhões (R\$ 26 bilhões) em 2002, sendo cerca de 79% referentes a serviços e os 21% restantes a *software* produto. O Gráfico 6 apresenta a evolução desse mercado nos quatro últimos anos.

Mercado Brasileiro

A pesquisa bienal elaborada pela Secretaria de Política de Informática/Ministério da Ciência e Tecnologia (Sepin/MCT) – *Qua-*

Gráfico 6
Evolução do Mercado Brasileiro de Software – 1999/2002
(Em US\$ Bilhões)



Fonte: Série Estudos Tecnologia da Informação – Software (*diversos anos*).

idade e Produtividade no Setor de Software Brasileiro (2001) – verificou que em 2000 havia, no Brasil, 10.713 estabelecimentos, e respectivos 158.353 empregados, com atividades relacionadas a *software*, estando aí incluídas a distribuição e editoração.

Já o faturamento total das empresas nacionais, de produtos e de serviços, em 2002, foi da ordem de R\$ 11,3 bilhões, de acordo com projeções realizadas pelo IEES.²⁴ Cabe observar que nesse cálculo estão incluídas empresas cujas receitas provêm, parcialmente, de outras atividades além do *software*, o que significa que a participação das empresas nacionais no mercado brasileiro é bem inferior a 50%. Há uma prevalência de pequenas e médias empresas e uma carência de empresas de maior porte. Do total da amostra utilizada pelo IEES, apenas 1,1% das empresas apresentou faturamento acima de R\$ 50 milhões (ver Tabela 4).

Ainda segundo projeções realizadas pelo IEES, estima-se que as empresas nacionais contem com cerca de 93.400 colaboradores, sendo que 51% delas possuem até nove colaboradores (ver Tabela 5).

Em relação à sua dispersão geográfica, há uma concentração na região Sudeste, onde se encontravam localizadas em 2003, segundo dados coletados pelo IEES, 64,4% das empresas brasileiras desenvolvedoras de *software*. Em seguida, destacava-se a região Sul, onde estavam cerca de 23,6% delas, enquanto as regiões Nordeste, Centro-Oeste e Norte concentravam 7,6%, 4,1% e 1% das empresas, respectivamente.

Tabela 4
Projeção do Faturamento das Empresas Nacionais de Software – 2002

FAIXA DE FATURAMENTO	NÚMERO DE EMPRESAS	%	FATURAMENTO (R\$ Milhões)	%
Até R\$ 1 Milhão	2.761	78,9	731,7	6,5
Acima de R\$ 1 Milhão até R\$ 5 Milhões	500	14,3	806,5	7,1
Acima de R\$ 5 Milhões até R\$ 10 Milhões	100	2,9	767,2	6,8
Acima de R\$ 10 Milhões até R\$ 20 Milhões	60	1,7	806,6	7,1
Acima de R\$ 20 Milhões até R\$ 50 Milhões	40	1,1	1.131,8	10,0
Acima de R\$ 50 Milhões até R\$ 150 Milhões	23	0,7	2.106,6	18,6
Acima de R\$ 150 Milhões até R\$ 300 Milhões	11	0,3	2.113,5	18,7
Acima de R\$ 300 Milhões	5	0,1	2.849,1	25,2
Total	3.500	100,0	11.313,0	100,0

Fonte: IEES.

²⁴O IEES possui uma base de dados com uma amostra representativa de cerca de três mil empresas nacionais desenvolvedoras de software.

Tabela 5

Projeção do Número de Colaboradores – 2002

FAIXA DE COLABORADORES	NÚMERO DE EMPRESAS	%	NÚMERO DE COLABORADORES (Milhares)	%
Até 9	1.791	51,2	10,0	10,7
De 10 até 24	1.200	34,3	18,1	19,4
De 25 até 99	300	8,6	14,7	15,8
De 100 até 199	130	3,7	16,8	18,0
De 200 até 399	53	1,5	14,3	15,3
De 400 até 999	22	0,6	12,8	13,7
Acima de 999	4	0,1	6,6	7,1
Total	3.500	100,0	93,4	100,0

Fonte: IEES.

A maior parte das empresas nacionais atua na prestação de serviços ou em desenvolvimento de aplicativos corporativos (de uso geral e/ou voltados para mercados verticais) e é majoritariamente dirigida para o mercado doméstico, sendo as exportações ainda pouco representativas nesse setor.

Um dos pontos fracos da indústria, principalmente sob a ótica da competitividade no mercado internacional, é a falta de empresas certificadas. No final de 2003, apenas 16 detinham certificação CMM níveis 2 e 3, não havendo nenhuma empresa brasileira com CMM níveis 4 ou 5.

A seguir serão apresentados separadamente os mercados de *software* produto e de serviços.²⁵

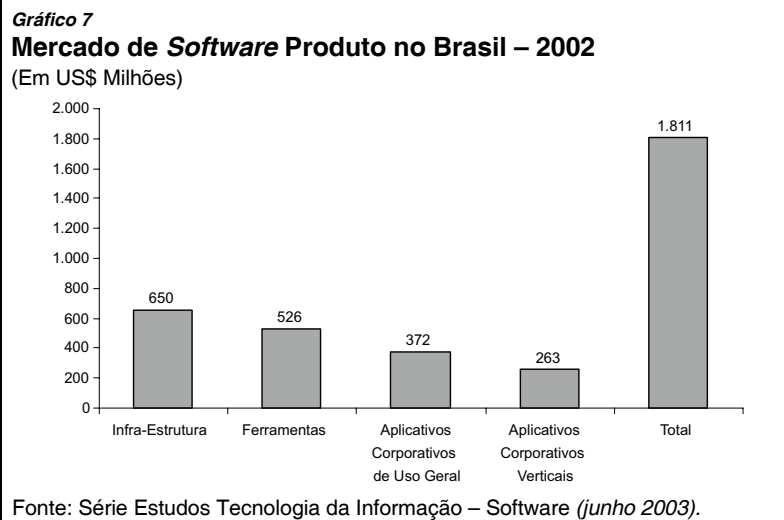
As vendas de licenças no Brasil somaram US\$ 1,81 bilhão (R\$ 5,41 bilhões) em 2002, representando uma queda nominal de 11,8% em dólar (e um crescimento de 18,2% em reais) em relação ao ano anterior. A baixa *performance* do setor é atribuída à queda dos investimentos em função do fraco desempenho da economia naquele ano, refletindo-se sobre os gastos em TI.

Os segmentos de infra-estrutura e de aplicativos corporativos – de uso geral e verticais – representavam, cada um, aproximadamente 35% das vendas totais, enquanto o segmento de ferramentas respondia pelos 29,1% restantes (ver Gráfico 7).

As empresas independentes de *software* (ISV) multinacionais concentraram a maior parte das vendas de licenças. Segundo a Computerworld, elas respondiam por cerca de 70% das vendas totais de *software* no país, seguidas dos fabricantes de *hardware*,

Setor de Software Produto

²⁵Os dados apresentados nesta seção foram retirados da revista Série Estudos Tecnologia da Informação – Software, de junho de 2003, publicada pela SHR Serviços de Assessoria & JP Consultores Associados, e das 100 Maiores de Informática 2003, da Computerworld, publicada pela IDG. Até o momento de encerramento deste artigo, as revistas de 2004, com dados referentes a 2003, não haviam sido publicadas.



com uma participação de 20%, e das empresas nacionais, com os 10% residuais.

A liderança das empresas multinacionais desenvolvedoras de *software* reflete a sua posição oligopolista em nível internacional, principalmente no segmento de infra-estrutura e ferramentas. A grande maioria, entretanto, não realiza nenhuma atividade de desenvolvimento no país. Seus principais concorrentes nesse segmento são os fabricantes de *hardware*, cuja atuação é explicada pelo fato de o *software* ser um componente estratégico para o funcionamento do equipamento.

As empresas nacionais atuam predominantemente no segmento de aplicativos, principalmente aqueles voltados para mercados verticais e de gestão empresarial. A participação de empresas nacionais nos segmentos de infra-estrutura e ferramentas limita-se aos produtos de segurança. A Figura 1 apresenta a atuação no mercado das principais empresas do setor, por segmento (as empresas de controle nacional estão destacadas em negrito).

A seguir será apresentada sucintamente a evolução de cada um dos três segmentos de *software* no Brasil em 2002.

Infra-Estrutura

As vendas de *software* de infra-estrutura somaram US\$ 650,2 milhões em 2002, mantendo-se estável em relação ao ano anterior (0,2%). Os sistemas operacionais – produto com maior participação no segmento (25,1%) – apresentaram o pior resultado, com uma retração de 10,7%. Em contrapartida, o *software* de segu-

Figura 1
Principais Empresas de Software Produto no Mercado Brasileiro

Empresas	Infra-Estrutura	Ferramentas	Aplicativos
BMC Software	████████████████████		
Borland		████████████████████	
CA	████████████████████		
Compuware		████████████████████	
Datasul			████████████████████
EMC	████████████████████		
HP	████████████████████		
IBM	████████████████████	████████████████████	
IFS			████████████████████
Interquadram			████████████████████
JD Edward			████████████████████
Legato	████████████████████		
Logocenter			████████████████████
Microsiga			████████████████████
Microsoft	████████████████████	████████████████████	████████████████████
MSA Infor		████████████████████	
Oracle		████████████████████	████████████████████
Peoplesoft			████████████████████
Peregrine	████████████████████		
Rational (IBM)		████████████████████	
RM Sistemas			████████████████████
SAP			████████████████████
Siebel			████████████████████
Software AG	████████████████████	████████████████████	████████████████████
SSI	████████████████████		
Sun Microsystems	████████████████████	████████████████████	
Sybase	████████████████████	████████████████████	████████████████████
Symantec	████████████████████		
Veritas	████████████████████		

Fonte: Série Estudos Tecnologia da Informação – Software (junho 2003).

rança teve uma significativa taxa de crescimento de 19,2%, representando cerca de 11,5% do mercado de infra-estrutura em 2002. Em que pese a desvalorização cambial ocorrida nesse ano, verifica-se uma contenção de novos investimentos em TI, sendo privilegiado o segmento de segurança.

As principais empresas atuantes no país são Microsoft, IBM, Computer Associates, HP, Novell, EMC, BMC Software e Oracle. Observa-se que prevalecem nesse segmento posições oligopolistas de grandes empresas multinacionais, enquanto a presença de empresas nacionais é pouco expressiva. Sua atuação está restrita ao *software* de segurança através das empresas Aker, Módulo, Scua e Open, entre outras.

A infra-estrutura é de extrema importância, uma vez que serve de base para o *software* aplicativo. Segundo pesquisa realizada pelo IEES, além do sistema operacional, o banco de dados é o *software* adicional mais exigido para o funcionamento dos produtos brasileiros. Na amostra pesquisada, 20,8% dos produtos utilizavam o banco de dados da Microsoft (SQL Server) e 18,3% o modelo da Oracle.

Os principais parceiros para comercialização do *software* de infra-estrutura no país são empresas (provedores) de serviços com forte presença na atividade de integração, como Atos Origin, EDS e T Systems.

O *software* livre tem se apresentado como uma alternativa a essa estrutura, sendo adotado por um número cada vez maior de empresas. Entre os setores que mais o utilizam no Brasil estão as grandes organizações do setor público e o varejo. Recentemente, iniciativas do governo federal têm apontado para uma crescente utilização do *software* livre.

Ferramentas

O mercado nacional de *software* para desenvolvimento foi de US\$ 526,6 milhões (R\$ 1,47 bilhão) em 2002, representando uma expansão de 4,2% (39,7% em real) em relação ao ano anterior. As empresas atuantes nesse mercado são majoritariamente de capital estrangeiro. Entre as maiores, destacam-se IBM, Microsoft, Oracle, Computer Associates, Borland e Novell, entre outras.

As ferramentas de desenvolvimento de objetos foram as que apresentaram maior taxa de crescimento (67,6% para valores em reais e 25% quando considerados valores em dólares), com uma participação de 9,8% nesse segmento. Já o *software* para modelagem de dados, produto com maior participação no segmento (39,1%), apresentou uma baixa *performance* quando comparado aos demais, com uma retração de 1,2% (e um crescimento de 32,4% em reais).

Os principais usuários de ferramentas de desenvolvimento no país têm sido as fábricas de *software* e as empresas desenvolvedoras de soluções de comércio eletrônico.

Aplicativos Corporativos

O mercado brasileiro de aplicativos corporativos apresentou uma retração de 35,5% das receitas em dólares (e 5,5% em reais) em 2002, caindo para US\$ 580,3 milhões, como reflexo da paralisação dos investimentos ocorrida. Desse total, os aplicativos corporativos de uso geral (interindústria), como ERP, CRM, SCM etc., somavam US\$ 371,7 milhões. As grandes marcas internacionais desses produtos, como SAP, Oracle e J.D. Edwards, têm forte atuação no mercado brasileiro, principalmente entre as empresas de maior porte e as multinacionais aqui estabelecidas. Um importante canal de comercialização desses produtos são os provedores de serviços.

O único mercado que pode ser considerado saturado nesse segmento é o de ERP para grandes empresas, sendo a SAP a líder do mercado. Os demais aplicativos corporativos de uso geral, bem como ERP para empresas de menor porte, apresentam baixos

índices de penetração no país, caracterizando um mercado ainda inexplorado. O baixo índice de digitalização do amplo universo de micro, pequenas e médias empresas (MPMEs) revela-se como um significativo mercado potencial.

Já os aplicativos corporativos voltados para o mercado vertical apresentaram um crescimento de 24% (66,3% em reais) em 2002, somando US\$ 262,6 milhões. Nesse segmento, destaca-se a forte presença de empresas nacionais, cuja vantagem competitiva baseia-se no alto conhecimento do negócio do cliente e do arcabouço legal e tributário do país. Em particular, quanto mais intenso for o ambiente regulatório do setor (por exemplo, setor bancário e telecomunicações), mais efetiva é essa vantagem competitiva.

O setor que apresentou maior participação na demanda foi o financeiro (18,3%), seguido do governo (16,5%). O primeiro destaca-se por seus altos investimentos em TI, impulsionados em grande parte pelas normas e exigências do Banco Central do Brasil, como, por exemplo, a adequação ao sistema de pagamentos brasileiro (SPB) e ao Acordo da Basiléia.

Os gastos em serviços de TI no país somaram em 2002 cerca de US\$ 6,64 bilhões (R\$ 20,68 bilhões), dentre os quais US\$ 1,72 bilhão referem-se a contratos de *outsourcing*. Todavia, houve uma retração da ordem de 8,7% em relação ao ano anterior.

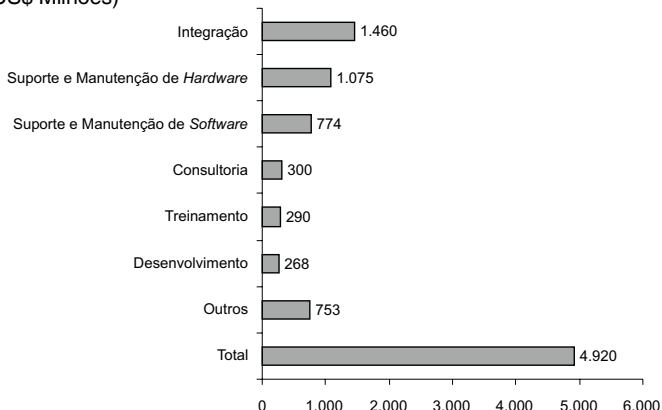
Serviços de Software

Os serviços profissionais somaram US\$ 4,92 bilhões, sendo que os gastos com estes serviços pela primeira vez superaram os gastos em *hardware* (US\$ 4,75 bilhões em 2002). O Gráfico 8 apresenta a participação de cada tipo de atividade.

Gráfico 8

Receitas de Serviços Profissionais no Brasil – 2002

(Em US\$ Milhões)



Fonte: Computerworld.

A maioria das empresas nacionais do setor é de pequeno e médio portes. Segundo a Computerworld, estima-se que existam de seis a sete mil empresas nacionais responsáveis por mais de 50% da receita advinda de prestação de serviços de TI. Uma particularidade dessas empresas diz respeito ao seu passivo trabalhista potencial, pois prevalece uma necessidade de contratação de mão-de-obra informal no setor para redução de custos.

Ao mesmo tempo, as grandes empresas multinacionais de serviços apresentam forte atuação no país e, de modo geral, não competem diretamente com as nacionais. Estas atuam como parceiras das grandes empresas, subcontratadas para a realização de projetos de maior porte. Além disso, sua atividade está concentrada em nichos de mercado, prestando serviços para clientes nacionais de pequeno e médio portes que não dispõem de recursos suficientes para contratar serviços das grandes marcas internacionais. A Figura 2 apresenta o posicionamento no mercado brasileiro dos principais prestadores de serviço (as empresas destacadas em negrito são de controle nacional).

Nos últimos anos observa-se o crescimento do número de fábricas de *software* no país (ver quadro a seguir). O êxito indiano na exportação desse tipo de serviços e a desvalorização da moeda, que tornou o custo da mão-de-obra nacional atrativo internacionalmente, têm sido os principais motivadores dessa expansão.

Fábrica de Software

A atividade de desenvolvimento de *software* pode ser dividida em quatro etapas, após a análise do negócio: especificação de requisitos; projeto e implementação; teste e validação; e manutenção. Tradicionalmente, a fase realizada pela fábrica de *software* é a implementação, recebendo por isso o nome de fábrica de programas.

A expressão fábrica de *software* está relacionada à tentativa de simular o processo fabril na atividade de desenvolvimento de *software*. A idéia é transformar a etapa da codificação do programa num processo padronizado, aumentando sua produtividade e eficiência.

As vantagens de uma fábrica de *software* são:

- maximizar a eficiência do programador;
- minimizar o tempo e o custo de programação; e
- manter um ambiente em que prevaleçam as melhores condições sob o ponto de vista da qualidade.

Um conceito de fábrica de *software* nasceu na GE, em 1968. A idéia era reduzir a variabilidade da produtividade da atividade de programação através da utilização de ferramentas e controles de gerenciamento. Ao mesmo tempo, na AT&T enfatizava-se o reúso sistemático de códigos ao construir programas. A Hitachi foi a primeira empresa a adotar a expressão fábrica de *software* em 1969.

Na década de 1990 houve um florescimento desse tipo de atividade. A busca pela redução de custos levou muitas empresas a terceirizar a codificação de seus programas recorrendo a fábricas de *software*. O modelo foi consagrado na Índia, onde as empresas beneficiaram-se do excesso de mão-de-obra qualificada e de baixo custo, apresentando elevadas taxas de crescimento nos últimos anos.

Algumas dessas empresas já dispõem de competência para a realização da etapa de desenvolvimento anterior à implementação, fazendo também a etapa de projeto, pelo que são chamadas fábricas de projetos.

Figura 2
Posicionamento dos Principais Prestadores de Serviços no Mercado Brasileiro

Empresas	Infra-Estrutura	Desenvolvimento de Software	Implementação de ERP, CRM etc.	Soluções para Mercados Verticais
Accenture				
Atos Origin				
Cimcorp				
Columbia Storage				
Convergys				
CPM				
CPqD				
DBA				
Dedalus				
Deloitte				
DTS Altran				
EDS				
Ernst & Young				
Eversystems				
Gennary & Peartree				
Politec				
Getronics				
GMC				
HP				
IBM				
InfoJBS				
Intec				
Integris				
Integra				
Itautec				
KPMG				
Lógica				
M13				
Medidata				
Perto				
Pimentel				
Plaut				
Polen				
Procomp				
ProcWork				
Prolan				
Promon Tecnologia				
SchulambergerSema				
Scopus				
Soluziona				
Sonda				
Stefanini				
T-Systems				
Unisys				
Vesta				
YKP				

Fonte: Série Estudos Tecnologia da Informação – Software (junho 2003).

A montagem de uma fábrica de *software* exige investimentos em infra-estrutura (*hardware*, estrutura de rede, ferramentas de desenvolvimento etc.), capacitação da equipe e certificação (CMM). Ao mesmo tempo, as margens desse negócio são pequenas, requerendo controle estrito de recursos (tempo e mão-de-obra) e estando sujeitas a pressões em função de variações cambiais, que afetam o custo da infra-estrutura. Dentre os principais clientes desse modelo de negócio, destacam-se os bancos, alguns possuindo empresas próprias para a realização desse serviço.

Além das fábricas vinculadas a bancos, destacam-se empresas de produtos, algumas delas grandes corporações internacio-

nais do setor, que criam suas próprias fábricas para fazer a localização (adequação às especificidades locais) e customização (adequação a um usuário) de seus produtos de *software*. Segundo dados veiculados na imprensa, a HP já gastou, em 2004, R\$ 1,8 milhão para ampliar o espaço das novas instalações de sua fábrica de *software* no Brasil. Figuram entre as maiores fábricas de *software* brasileiras: CPM, DTS, eDablio, HP, IBM, Itaotec Philco, RM, SI Brasil, Stefanini, TBA e Vesta.

A oportunidade de prestação desse tipo de serviço no mercado internacional, especialmente nos Estados Unidos, muito bem explorada por países como a Índia, tem despertado grandes expectativas. Entretanto, esse tipo de serviço não cria maiores vínculos entre prestador e cliente e, uma vez obedecidos os pré-requisitos de qualidade (casos de sucesso, certificação etc.), o verdadeiro determinante da contratação é o preço, extremamente sensível a variações cambiais.

Por fim, o mercado de *outsourcing*, seguindo a tendência mundial, tem apresentado recorrentes taxas de crescimento. Observou-se uma expansão de 9,3% entre 2001 e 2002 (44,3% quando considerados os valores em reais). Segundo a IDC Brasil, espera-se que esse mercado atinja R\$ 3,4 bilhões em 2004.

Ainda segundo a IDC Brasil, os grandes fornecedores detêm a maior parte do mercado, com uma participação de 45%, destacando-se IBM, EDS, Atos Origin, Accenture, CPM e DBA, entre outras.

Os serviços de *outsourcing* realizados no país normalmente seguem o modelo convencional, isto é, a terceirização de serviços de infra-estrutura e de gestão de aplicativos. Os contratos de BPO ainda são pouco expressivos.

Software Livre

Embora ainda muito concentrada nos Estados Unidos, berço do movimento, a utilização de *software* livre vem sendo difundida em todo o mundo, como pode ser inferido a partir de estatísticas que fornecem o número de distribuições (empresas distribuidoras) de *software* livre. Uma delas, disponível no site <http://www.distrowatch.com>, diz que no início de agosto de 2004 havia 322 distribuições no mundo, das quais 32 descontinuadas. A sua presença nos países indica que quase um quinto das distribuições está nos Estados Unidos (59), vindo a seguir Alemanha (21), Canadá e França (19 cada), Espanha (16) e, empatados, Brasil, Itália e Japão (11 cada).

A adesão de grandes corporações internacionais ao *software* livre deu-se ao final da década de 1990, inclusive patrocinando

o seu desenvolvimento. A primeira empresa a fazê-lo foi a IBM, que em 1999 criou o *Linux Technology Center* e é hoje a maior vendedora de Linux e respectivos serviços de suporte do mundo, segundo a *Business Week*. Essa decisão foi inesperada pela aparente contradição entre a forma de licenciamento proprietário que praticavam e a liberdade do *software*. Todavia, uma análise mais apurada revela uma estratégia muito bem construída de redução de custos, ampliação de participação em mercados – o que pode incluir a penetração em mercados monopolistas – e mudança no modelo de negócios.

Vale observar a oportunidade que o *software* livre representa para os governos, muito além da redução de custos, pelo seu potencial ganho em transparência, segurança, formação de mão-de-obra especializada e, principalmente, aplicações sociais. Entretanto, tudo isso precisa ser feito com a inclusão da indústria, pois cabe ao governo o papel de formulador de políticas públicas, sociais e também industriais.

A idéia do trabalho cooperativo vem permeando a vida acadêmica há muitos anos. É assim com o desenvolvimento científico, em que pesquisas são divulgadas e submetidas livremente a toda a comunidade, a qual as estuda, oferece suas críticas e utiliza seus resultados em novas pesquisas. Particularizando para o *software*, foi nesse ambiente que se criaram e desenvolveram diversos padrões da internet, como o HTML, a linguagem Perl e o protocolo TCP/IP.

Sua história, de alguma forma, está ligada ao sistema operacional Unix, que teve as raízes de seu desenvolvimento na década de 1960, em um trabalho conjunto entre o AT&T Bell Labs, o MIT e a GE. O projeto passou a ser responsabilidade da AT&T anos mais tarde, sendo reescrito para uma máquina da Digital e, em 1973, para a linguagem C. Isso permitia que fosse facilmente portado, isto é, modificado, de forma a poder trabalhar em outras máquinas do mesmo fabricante ou de outros.

O Unix foi disponibilizado para um grande número de universidades a preço simbólico, e também para a Universidade da Califórnia, em Berkeley, a qual recebeu da empresa o código-fonte, o que lhe permitiu estudar o Unix e passar a desenvolver a sua própria versão, conhecida como *Berkeley software distribution* (BSD-Unix). Ao mesmo tempo, a AT&T vendia versões comerciais do produto, sem o código-fonte, para o mercado corporativo. O conflito entre a distribuição livre do BSD-Unix e a distribuição proprietária do Unix da AT&T foi inevitável.

Histórico do Movimento

Foi nesse ambiente que, em 1984, nasceu o movimento do *software* livre, com a saída de Richard Stallman do Laboratório de Inteligência Artificial do MIT, onde trabalhava desde 1971. Durante vários anos ele havia participado do desenvolvimento do sistema operacional de um computador utilizado pelo Laboratório. Com a substituição da máquina por outra mais moderna, cujo sistema operacional era proprietário, Stallman passou a defrontar-se com restrições à sua liberdade de *hacker*.²⁶ Não havia acesso ao código-fonte do sistema, as modificações de que precisava tinham que ser solicitadas ao fornecedor do *software*, este não podia ser compartilhado com terceiros etc. Com a criação do novo movimento, Stallman buscava restabelecer a vivência em comunidade, a cooperação e a liberdade que experimentara no Laboratório, colocando suas habilidades técnicas a serviço de todos e, naturalmente, opondo-se à lógica da propriedade intelectual sobre o *software*.

Seu propósito era o desenvolvimento de um sistema operacional próprio, portátil, compatível com o Unix, de forma a poder substituí-lo facilmente nas instalações dos usuários. Deu ao sistema o nome de GNU, formado pelas iniciais de *Gnu is Not Unix*, o qual deveria ter, além do núcleo principal (*kernel*), diversos outros módulos como processadores de comando, montadores, compiladores, interpretadores, editores de texto etc. Assim sendo, Stallman lançou-se ao desenvolvimento do compilador GCC e do editor GNU Emacs, que despertaram o interesse de outras pessoas que aderiram ao projeto GNU, dando origem, em 1985, à Free Software Foundation.

Para Stallman, todo *software* deve ser livre (*free*), o que implica que todos os usuários sejam livres para:

- usar o programa, qualquer que seja o propósito;
- modificar o programa de forma a atender às suas necessidades;
- distribuir cópias do programa, gratuitamente ou não; e
- distribuir cópias modificadas do programa, de forma a permitir que outros usufruam das modificações introduzidas.

A liberdade de modificação implica o acesso ao código-fonte do programa, pois só assim ele pode ser estudado e modificado.

Todos os *softwares* GNU são protegidos por uma licença própria – a General Public License (GPL) –, que utiliza a legislação da propriedade intelectual de forma reversa (*copyleft*)²⁷ para garantir que ninguém possa restringir a liberdade do *software*. Assim, a adição ou combinação de qualquer *software* proprietário a *software* coberto pela licença GPL resulta em um produto forçosamente GPL. Daí dizer-se que há “contaminação” do novo *software* pela licença GPL.

²⁶ O termo significava, e significa ainda para a comunidade do *software* livre, alguém que ama programar e se diverte ao exercitar suas habilidades. O sentido normalmente atribuído à palavra e difundido pelos meios de comunicação – alguém que burla a segurança – não é aceito por aquela comunidade.

²⁷ Em oposição a copyright.

A única exceção à regra do licenciamento GPL, no sistema GNU, é a biblioteca C do sistema – GNU C *library* –, que é protegida por uma licença denominada “*Lesser*”, ou Library General Public License (LGPL), a qual permite que aplicações proprietárias sejam compiladas para operarem no sistema GNU, o que acarreta serem conectadas a essa biblioteca sem perderem sua característica original. Tal estratégia tem permitido a difusão do sistema GNU em muito maior escala do que a que seria atingida se o sistema ficasse restrito ao universo das aplicações livres.

O sistema GNU cresceu muito, não somente com o trabalho dos colaboradores da Free Software Foundation, mas também com a incorporação de programas livres de outras origens. Diversos programas do sistema GNU tornaram-se populares devido à sua qualidade e foram portados pelos usuários para vários sistemas. Ao chegar à década de 1990, o sistema GNU estava quase todo desenvolvido, à exceção do *kernel*, questão que acabou sendo resolvida com o surgimento do Linux.

Em 1991, um estudante da Universidade de Helsinque, Linus Torvald, anunciou na internet que estava trabalhando no desenvolvimento de um *kernel* de sistema operacional para o PC 386 AT, para o qual já havia portado dois programas GNU – o GCC e o BASH (*Shell*, interface de linha de comando que permite a comunicação do usuário com o sistema). Era sua intenção que o novo *kernel* substituísse o Minix, versão de Unix desenvolvida para microcomputadores, com o qual não estava satisfeito. Três anos mais tarde, o *kernel*, nomeado Linux, teve sua versão 1.0 liberada para uso, sendo protegido pela licença GPL.

O Linux foi adotado como *kernel* do sistema GNU, de forma que qualquer referência ao sistema operacional significa uma referência ao sistema GNU/Linux, que, de acordo com a Free Software Foundation, possui cerca de 10 milhões de usuários ao redor do mundo.

Em 1998, surgiu na Califórnia uma dissidência do movimento do *software* livre liderada por Eric Raymond, que, como ex-participante da Free Software Foundation, havia se tornado muito conhecido na comunidade do *software* livre. A dissidência adotou o conceito de *open source* (fonte aberta) em detrimento do de *software* livre, buscando conciliar as liberdades de uso, modificação e cópia com os interesses das empresas.

A iniciativa deu origem à Open Source Initiative, a qual anunciou, ao final de 1999, a criação de um certificado OSI de *open source*. Sua definição do que seja um *software* que mereça essa qualificação compreende o atendimento de alguns requisitos:

- a redistribuição é livre, não podendo ser restringida a venda ou cessão do *software* enquanto componente de uma distribuição que reúna produtos de origens distintas;

- o código-fonte deve ser aberto;
- as modificações e os trabalhos derivados podem ser redistribuídos nos termos da licença do *software* original;
- deve ser garantida a integridade do código-fonte do autor, sendo as modificações explicitadas ou o original facilmente reconstruído;
- não pode haver discriminação quanto a pessoas, grupos ou usos;
- a licença não pode contaminar outros *softwares* distribuídos em conjunto; e
- o licenciamento deve ser neutro quanto ao tipo de tecnologia utilizado.

Em 2000, foi fundada a Open Source Development Labs, dedicada à disponibilização, nos Estados Unidos e no Japão, de recursos modernos e profissionais de computação e teste a desenvolvedores do Linux. Seu objetivo é direcionar e acelerar o desenvolvimento daquele *kernel* para uso corporativo, para o que contribui também com a participação dos usuários junto à comunidade desenvolvedora. Algumas grandes empresas como IBM, HP, CA, Intel e NEC participaram da fundação da Open Source Development Labs. Atualmente, o número de membros famosos atinge algumas dezenas, destacando-se Alcatel, AMD, Cisco, Ericsson, Fujitsu, Hitachi, Mitsubishi, Nokia, Novell, Sun, NTT, Toshiba e Unilever, além de grandes distribuições Linux como a Red Hat.

O trabalho de desenvolvimento e evolução do Linux continua sendo liderado pelo próprio Linus Torvald, o qual é patrocinado pelo Open Source Development Labs.

Modelo de Desenvolvimento de Software

Nesta subseção não será feita distinção entre *software* livre e *open source*, uma vez que os modelos de desenvolvimento são idênticos, sendo utilizada a nomenclatura *software* livre para designar uma e outra coisa.

Um novo projeto de *software* livre tem início com a publicação, pelo autor, do seu código-fonte em um *site* na internet. A partir desse momento podem ocorrer adesões voluntárias de desenvolvedores ao projeto, cujas contribuições são normalmente efetuadas por meio de listas de discussões. Em geral, existem duas listas para cada projeto, uma dedicada ao desenvolvimento e outra ao suporte, com problemas e sugestões.

Todo projeto tem pelo menos um mantenedor – líder do projeto –, o qual é responsável pela incorporação das modificações propostas ao código-fonte e também decide se uma versão em

desenvolvimento já está suficientemente estável para ser transformada em versão para distribuição. Comumente, a função de mantenedor é assumida pelo autor ao publicar o *software* na internet.

Alguns projetos, especialmente os maiores, possuem uma espécie de colegiado de mantenedores, havendo, entretanto, diferentes formas de relacionamento com a comunidade de desenvolvedores. Em alguns casos, existe uma estrutura marcadamente hierárquica, tanto de acompanhamento dos “novatos” quanto de decisão em relação às modificações aceitas. Já em outros projetos a estrutura é mais democrática, sendo as decisões tomadas por consenso.

Não existe “chefia”, assim como não há pagamento pelo trabalho voluntário. Existem lideranças de projeto, baseadas no mérito demonstrado, e o reconhecimento das contribuições prestadas por um dado colaborador, através do prestígio que granjeia entre seus pares. Em alguns projetos, por exemplo, o nome do colaborador é explicitamente associado à modificação incorporada ao código-fonte.

O *software* livre normalmente não possui um plano de evolução definido. No caso do *software* proprietário, existe sempre um planejamento que norteia a vida do produto e é imposto como balizador para o seu desenvolvimento. No caso do *software* livre, a evolução vai sendo construída pelos próprios desenvolvedores ao longo de seu trabalho, assim como pelo mantenedor ao aprovar as modificações.

Não há prazos condicionando o final da etapa de desenvolvimento e a sua liberação para o mercado. Isso permite que o *software* seja convenientemente testado e depurado, com o concurso de um grande número de pessoas, o que tende a conferir maior confiabilidade ao produto. Como não há pressões comerciais, qualquer modificação ou nova versão só é liberada para uso quando considerada estável. Por outro lado, por não haver compromissos com prazos, não pode ser feita qualquer previsão sobre a disponibilidade de versões ou alterações.

Sobre esse aspecto, merece destaque o Linux. Sendo realizado, hoje, por um grande número de colaboradores, seu desenvolvimento ocorre com uma velocidade e um grau de qualidade superiores aos normalmente apresentados por uma empresa de *software*.

Atualmente, existem na internet milhares de projetos de *software* livre. Entretanto, grande parte deles não conseguiu a adesão da comunidade. É preciso algum tempo para que um projeto atinja um número suficiente de colaboradores e um certo dinamismo, baseado em desenvolvimento e uso, para que possa ser considerado um caso de sucesso. Este é marcado pelo surgimento de todo um conjunto de atividades externas ao projeto, que são, ao mesmo

tempo, facilitadores e indicadores da sua penetração – serviços de consultoria, distribuição, suporte, treinamento, edição de livros sobre o *software* etc.

Alguns fatores influenciam a adesão dos desenvolvedores a determinados projetos, como, por exemplo, a vinculação a empresas faz com que os projetos sejam encarados com desconfiança, pois a comunidade reluta em contribuir para que eles possam ser apropriados por terceiros. Isso faz com que as empresas que desejem participar de um desenvolvimento normalmente o façam através dos endereços eletrônicos particulares de seus funcionários.

Para que um projeto cresça e atinja massa crítica é necessário que todas as suas regras de gestão sejam prévia e claramente definidas. Além disso, ele precisa ser considerado tecnicamente desafiador, para que os colaboradores voluntários se sintam motivados a superar as dificuldades que ele apresenta e a propor soluções elegantes para os problemas. Assim é que projetos considerados maduros deixam de exercer sua atratividade e acabam por ver diminuir o número de colaboradores.

Exerce também uma forte influência sobre a motivação dos colaboradores o desejo de compartilhar suas habilidades com outros desenvolvedores de *software*, participando de um trabalho cooperativo, diferentemente do que acontece na indústria do *software* proprietário, ancorada na competição.

O mantenedor tem também um papel importantíssimo no sucesso ou não de um projeto, pois é a ele que cabe motivar a comunidade, administrar conflitos de interesses e exercer sua liderança de forma a não provocar dissidências, ou seja, o surgimento de projetos alternativos que, freqüentemente, dão origem a soluções conflitantes.

Em 2002, a Universidade de Maastricht, na Holanda, conduziu uma pesquisa que objetivava, entre outras coisas, conhecer a motivação dos colaboradores de projetos de *software* livre e *open source* (tratados em conjunto), suas características pessoais e sua percepção sobre esse tipo de *software* versus o *software* proprietário. O questionário foi colocado na internet e divulgado em diversos *sites* de desenvolvimento desses projetos, sem restrições ao número de entrevistados ou à sua origem, tendo sido respondido em dois meses por 2.784 pessoas, o que foi considerado uma boa amostra.

A pesquisa verificou que 70% dos desenvolvedores entrevistados possuíam graduação universitária e que cerca de 83% estavam empregados no setor de TI ou eram ligados a essa área na universidade. Os engenheiros de *software* respondiam pelo maior grupo profissional individual, perfazendo um terço da amostra, enquanto os estudantes de TI vinham em segundo lugar, com cerca de

16% da amostra. Contudo, os engenheiros de *software* e os programadores eram os que dedicavam mais de 20 horas semanais ao desenvolvimento de *software* livre, ambos com 72%, ao passo que apenas 17% dos estudantes destinavam mais de 20 horas semanais a essa atividade.

Essas constatações jogam por terra a visão de que o desenvolvimento de *software* livre é uma atividade conduzida por pessoas não suficientemente qualificadas. Foi demonstrado que é uma tarefa realizada majoritariamente por profissionais especializados, que a exercem de forma livre e voluntária, tendo por principal objetivo o aperfeiçoamento de suas habilidades em uma relação de troca com outros profissionais.

Mais da metade dos entrevistados pela pesquisa (51%) tinha sua principal fonte de renda no desenvolvimento de *software*, enquanto 46% eram pagos para administrar, suportar ou desenvolver *software* livre, o que é um indicador da participação das empresas no desenvolvimento desse tipo de *software*. Ao utilizá-los como base para seus fornecimentos, participar do desenvolvimento foi uma forma encontrada por empresas de TI para garantir e influenciar a evolução dos produtos.

A pesquisa encontrou uma larga maioria de desenvolvedores fisicamente localizados na Europa e nos Estados Unidos/Canadá – cerca de 80% dos entrevistados –, o que qualitativamente confirmou resultados de outras pesquisas anteriores.

Outro fato importante é que 48% dos entrevistados disseram-se ligados ao movimento do *software* livre, enquanto outros 33% afirmaram fazer parte da comunidade *open source*. Para os restantes, cerca de um quinto do total, isso foi indiferente.

Os primeiros programas eram gratuitos e acompanhavam os computadores, que praticamente concentravam o valor dos sistemas. Foi com a separação entre *hardware* e *software* e a crescente importância do *software* dentro dos sistemas que ele passou a ser cobrado, o que, juntamente com a padronização das máquinas, motivou a criação de empresas fornecedoras independentes. A proteção da propriedade intelectual sobre o *software* veio em seguida, de forma a garantir os direitos das empresas e suas vantagens competitivas.

Com a microinformática, dois tipos de *software* diferentes surgiram, o *freeware* e o *shareware*, ambos distribuídos gratuitamente e apenas sob a forma de código executável. O primeiro pode ser usado por um período de tempo, definido pelo proprietário, após o

Tipos de Software e Liberdade

que deixa de funcionar ou o faz de maneira limitada, sendo usado como demonstração, por exemplo. Já o segundo pode ser usado, e eventualmente distribuído, sem restrições de prazo, estando normalmente ligado a estratégias de *marketing* como a rápida ocupação de um mercado. Nenhum dos dois tipos merece a denominação de *software* livre.

Tampouco pode ser caracterizado como livre o *software* de domínio público, pois pode ser livremente obtido. Porém, por não ser protegido por qualquer licenciamento, nada impede que a partir dele seja criada uma versão modificada distribuída sob a forma proprietária.

Um outro tipo de *software*, de origem mais recente, é o *shared source*, no qual parte do código-fonte é liberado, somente para leitura, para alguns usuários selecionados. Há restrições à liberdade de cópia ou distribuição, e a modificação é impedida, devendo ser solicitada à proprietária do *software*. Portanto, não se caracteriza como sendo livre.

O conceito de *software* livre modifica a propriedade do *software*, pois pressupõe a liberdade de usar, modificar, copiar e distribuir o programa. O código-fonte é distribuído junto com o produto, por ser indispensável à liberdade de modificação. É importante observar que a simples entrega do código aberto não caracteriza um produto de *software* como sendo livre, pois o fornecimento pode ser protegido por um licenciamento proprietário restringindo a liberdade de distribuição, por exemplo.

O *software* livre não implica gratuidade. Porém, o fato de poder ser facilmente obtido a partir de algum *site* na internet regula o montante que pode ser cobrado pela sua distribuição, ficando esse valor restrito ao preço da mídia física e da documentação agregada. Isso não impede que outros tipos de *software* proprietário sejam comercializados em conjunto com o *software* livre, bem como serviços de consultoria, integração, suporte e treinamento, todos normalmente pagos.

Produtos de Software Livre

O fato de a maioria dos colaboradores ser constituída por pessoas ligadas ao desenvolvimento de *software*, em empresas ou universidades, ou seja, pessoas que lidam cotidianamente com a tecnologia, ocasiona um dos conhecidos pontos fracos do *software* livre – a interface homem-máquina. A motivação que move os desenvolvedores normalmente prende-se ao desenvolvimento de habilidades e à solução de problemas e não à criação de produtos “amigáveis”. Para isso contribui também o fato de os usuários se fazerem presentes somente nas listas de discussão voltadas para o suporte. Como consequência, tem-se que os maiores casos de

sucesso de *software* livre são representados por sistemas operacionais e produtos para servidores, mais ligados às máquinas e menos aos usuários.

O aprimoramento das interfaces e dos aplicativos é uma preocupação mais recente e está na ordem do dia, com a prioridade que alguns governos vêm concedendo à adoção de produtos para estações de usuários baseados em *software* livre, como é o caso do Mandrake na França, do Gnome na Espanha (Extremadura) e do OpenOffice nos telecentros de São Paulo.

A seguir são apresentados alguns casos de *software* livre de sucesso.

Apache

Foi criado na primeira metade da década de 1990 no National Center for Supercomputing Applications. Tendo seu código-fonte publicado, passou a receber um grande número de contribuições de *webmasters*, que sugeriam modificações (*patches*), o que acabou dando origem ao seu nome – Apache. Um pequeno grupo desses *webmasters* fundou o Apache Group, que em 1999 transformou-se na Apache Software Foundation, atual responsável pela sua manutenção.

O Apache é o servidor de Web em plataforma Linux ou Unix mais popular do mercado, considerado maduro e comoditizado, do qual detém uma participação superior a 60%. Vale observar que existem hoje diversas variações do Unix, incompatíveis entre si, desenvolvidas pelos fabricantes de *hardware* para as suas máquinas específicas. Assim, a IBM possui o AIX, a HP o HP-UX, a Sun o Solaris, a Silicon Graphics o Irix etc.

Sua forma de licenciamento permite que sejam introduzidas modificações no *software* e que este seja distribuído apenas sob a forma executável. Porém, a versão modificada não pode ser distribuída com o nome Apache, possibilitando que diversas empresas utilizem o *software* como parte da infra-estrutura sobre a qual podem ser construídos seus sistemas proprietários. É o caso da IBM e da BEA.

FreeBSD, OpenBSD e NetBSD

Versões abertas do Unix, foram criadas a partir de uma versão comercial completa do BSD-Unix para a plataforma Intel produzida na década de 1990 por desenvolvedores oriundos da Universidade da Califórnia. São utilizadas em grande número dos servidores de internet.

Seu modelo de licenciamento deriva do BSD, permitindo que modificações sejam feitas ao original e tornem-se proprietárias, isto é, possam ser distribuídas apenas sob a forma executável e cobradas.

Berkeley Internet Name Domain (BIND)

Implementação de protocolos que provê o serviço de *domain name system* (DNS) para toda a internet, está presente em mais de 75% dos servidores de DNS, de acordo com o Internet Software Consortium, organização sem fins lucrativos responsável pela sua manutenção e à qual estão associadas, entre outras, a IBM, a HP, a Sun e a Apple. Sua forma de licenciamento é muito semelhante à do BSD.

Gnome

É um ambiente para estações de usuários com interface gráfica e plataforma de desenvolvimento compatível com GNU/Linux, Solaris, HP-UX, BSD e Darwin (também um projeto BSD). Seu desenvolvimento está a cargo da Gnome Foundation, patrocinada, entre outras, por HP, IBM, Sun, Novell, Free Software Foundation e as distribuições Linux Red Hat, Mandrake e Debian. O Gnome faz parte do projeto GNU.

OpenOffice

É a mais difundida suíte de escritórios para microcomputadores do tipo estação de usuário alternativa ao Office da Microsoft. Possui processador de textos, planilha, gerenciador de apresentações, programa de desenho e ferramentas de banco de dados, trabalha com diversos formatos de arquivos e possui elevada compatibilidade com o Office. É importante observar que essa compatibilidade é muito difícil de ser atingida, pois, por ser proprietário, o formato dos arquivos Office não está disponível, dificultando a sua recriação no *software* livre, especialmente no que se refere às funcionalidades mais complexas. O OpenOffice funciona atualmente nas plataformas Solaris, Linux e Windows.

Seu código-fonte foi liberado para a comunidade (OpenOffice.org) pela Sun Microsystems a partir da versão 6.0 do StarOffice (versões anteriores eram proprietárias, oriundas da alemã StarDivision, adquirida pela Sun). Algumas funcionalidades foram retiradas do produto pela Sun antes de colocá-lo na OpenOffice.org, em particular aquelas baseadas em código de terceiros e sobre as quais existem cláusulas de confidencialidade. De forma análoga, o OpenOffice é base para as novas versões do StarOffice, o qual contém funcionalidades adicionadas pela Sun ao código original. O OpenOffice.org é patrocinado pela Sun.

O OpenOffice é distribuído sob duas formas de licenciamento, LGPL e Sun Industry Standards Source License (SISSL), esta última permitindo a não distribuição do código-fonte das modificações realizadas e o uso de formas de licenciamento proprietário.

Linux

O Linux é um *kernel* de sistema operacional ao qual podem estar associados programas GNU, caso em que merece a denominação GNU/Linux, ou de outras procedências, o que freqüentemente acontece nas distribuições.

Disponibilizado para a comunidade em 1991, sua evolução tem sido fantástica – a última versão liberada para uso, a 2.6, possui mais de quatro milhões de linhas de código. Tendo sido inicialmente desenvolvido para o Intel 386, foi modificado para facilitar o trabalho de portá-lo para outros processadores. Hoje seu desenvolvimento está centrado em Intel de 32 bits, porém com grupos específicos de colaboradores para outros processadores – Power PC (IBM), PA-Risc (HP) e Sparc (Sun). Sua arquitetura inicial, monolítica, foi tornada modular, o que facilita imensamente o trabalho de desenvolvimento. A modularidade torna possível também o trabalho de um grande número de colaboradores em paralelo.

Diversas grandes empresas mantêm grupos de desenvolvimento dedicados a contribuir para a comunidade Linux, pois essa é uma forma de ajudar a construção de um plano de evolução do produto, além de garantir a sua qualidade, imprescindível enquanto parte da infra-estrutura que essas empresas fornecem e suportam. A propósito, o trabalho de portar o Linux para arquiteturas proprietárias, baseadas nos processadores específicos das grandes fabricantes de *hardware* acima citados, foi realizado no seio dessas próprias empresas.

O Linux hoje está disponível para uso em estações de usuários, servidores, *clusters* (conjuntos de computadores interconectados, desenvolvidos como alternativa mais barata para os supercomputadores) e em versões “embarcadas”. O mercado de *software* embarcado é fragmentado e “invisível”, pois o *software* está presente, como componente, em um grande número de aparelhos. O Linux, da mesma forma que outros *softwares* livres, vem sendo adaptado para uma série dessas utilizações, tendência que visa não somente à utilização de sistemas estáveis e de qualidade reconhecida, como também à redução de custos pelo não pagamento de *royalties*. A adaptação para uso embarcado pressupõe conhecimentos específicos da utilização, razão por que é grande a presença de empresas dos respectivos setores no desenvolvimento do Linux.

Tendo em vista a sua origem inspirada no Unix, o Linux pode substituí-lo em uma instalação com relativa facilidade, reque-

rendo pequeno esforço de adaptação dos usuários. Segundo a IDC, em 2002 o Linux já teria suplantado a presença do Unix em servidores, como visto na seção anterior. Na região Ásia-Pacífico, de acordo com a Gartner Inc., 42% das corporações possuem pelo menos um servidor com Linux.

No Brasil, uma pesquisa da E-Consulting divulgada ao final do primeiro trimestre de 2004 constatou a presença do Linux em pelo menos um servidor, em 78% das médias e grandes empresas. No caso dos bancos, de acordo com a Federação Brasileira de Bancos (Febraban), o número dos que adotam Linux em servidores e em ambientes de escritório é de 42%.

Quanto às distribuições do Linux, é importante observar que modificações realizadas sob encomenda de determinados usuários não são incorporadas ao código do *software* livre em desenvolvimento, precisando ser atualizadas quando da instalação de novas versões.

As modificações introduzidas e não incorporadas ao produto, assim como módulos proprietários adicionados ao *kernel*, fazem com que possam haver no mercado diversos Linux diferentes entre si, muitas vezes não sendo possível intercambiar aplicativos entre eles.

Modelos de Negócio

O desenvolvimento compartilhado em uma comunidade aberta, como é o caso do *software* livre, não é adequado ao caso de uma inovação que possa trazer um diferencial competitivo a seu criador, pelo fato de todos os segredos do produto imediatamente tornarem-se públicos e, portanto, conhecidos dos produtores concorrentes.

O grande benefício trazido pelo *software* livre para as empresas desenvolvedoras é o da redução de custos. No modelo tradicional de direitos autorais, a criação, a evolução e a manutenção de qualquer produto de *software* configuram um investimento que precisa ser amortizado pelas vendas de licenças do produto. Por outro lado, esse mesmo desenvolvimento, quando feito por uma comunidade livre, proporciona uma diluição dos custos pela rede, reduzindo o preço final do produto.

O *software* livre representa uma opção no sentido da padronização, pois tem como diretriz a adoção de padrões abertos – plataformas de *hardware*, *software*, protocolos de comunicação, linguagens etc. –, o que propicia fácil portabilidade dos produtos e, portanto, maior poder de disseminação no mercado. Essa postura é oposta à dos padrões proprietários, que atam os usuários a fornece-

dores únicos, ensejando a prática de preços elevados. No caso dos sistemas operacionais, por exemplo, a grande maioria dos usuários não está disposta a pagar mais por diferenciações. A padronização interessa aos usuários também pela possibilidade de interoperabilidade e pela redução de risco de novos investimentos em aplicativos. Do lado da indústria de *software*, a disseminação do *software* livre como plataforma alarga o mercado dos aplicativos e reduz a necessidade de pessoal de suporte especializado em plataformas proprietárias.

Assim sendo, produtos comoditizados ou a caminho, em que a competição está baseada em custos, são naturais candidatos a serem desenvolvidos como *software* livre. Isso acarreta a transformação de um negócio centrado na venda de licenças de produtos em um negócio voltado para a venda de soluções integradas – *software* livre associado a outros produtos proprietários – e de serviços.

São também desenvolvimentos típicos de *software* livre os produtos maduros, para os quais os usuários não estão dispostos a pagar um preço-prêmio por funcionalidades adicionais. Nesse caso, dificilmente uma empresa conseguirá amortizar os investimentos constantes necessários à sua manutenção. Uma saída adotada é o compartilhamento do projeto com a comunidade, diluindo os custos de desenvolvimento e permitindo à empresa fixar-se em uma faixa superior do produto, no qual a diferenciação por novas funcionalidades ainda é viável para alguns tipos de clientes. Essa estratégia permite à empresa continuar disputando a faixa inferior do mercado através da concentração em serviços associados ao produto e/ou promoção da venda de *hardware* próprio.

Além disso, no caso de um produto cuja participação no mercado é muito pequena, acontece de a empresa proprietária disponibilizar o código-fonte para a comunidade e transformar a sua manutenção em projeto de *software* livre. Isso viabiliza que a empresa continue no mercado, aumentando a participação do produto e fortalecendo sua marca, a ser explorada em outros produtos.

Existem casos, também, de projetos novos e com alta complexidade, cuja empresa criadora não tem recursos para completar o seu desenvolvimento. Disponibiliza o código-fonte para a comunidade, com quem passa a compartilhar o trabalho de desenvolvimento e manutenção. Seu negócio passa a ser, principalmente, o de suporte ao produto, serviço no qual terá inegável vantagem competitiva.

O *software* livre permite também que diversas novas empresas sejam criadas, voltadas para a distribuição e a prestação de serviços vinculados ao produto distribuído. Incluem-se aqui modificações para adaptação a necessidades específicas do cliente, instalação, manutenção, suporte e treinamento. Outra importante atri-

buição dessas empresas é a elaboração de documentação para os usuários, coisa que praticamente inexistente nos *sites* de *software* livre.

De forma geral, o *software* livre tem promovido o deslocamento do foco do negócio: enquanto no *software* proprietário ele está no fornecedor e na promoção comercial do produto, no *software* livre ele se encontra no usuário e nos serviços agregados.

Penetração em Mercados Proprietários

Foi visto que a ampla disseminação de padrões proprietários em um mercado pode dar origem a monopólios *de facto*, sendo a principal barreira à entrada de novos concorrentes constituída pela interoperabilidade estabelecida entre as várias cópias do *software* disseminadas pelo mercado. O valor do investimento necessário para vencer uma tal barreira é muito alto, pois pressupõe o desenvolvimento de uma nova tecnologia capaz de suplantá-la existente, além de outros custos associados ao convencimento do usuário sobre o valor superior do novo produto. Esse trabalho de convencimento inclui publicidade do produto e do fornecedor, criação de canais de distribuição e treinamento de usuários, entre outros.

Tudo isso exige tempo, inclusive para que seja criada uma infra-estrutura de serviços de apoio ao uso do *software*, sem o que o produto não consegue atingir um nível de presença no mercado que impulse o seu desenvolvimento. Alcançado esse ponto, o novo *software* passa a valer-se também do efeito de rede. Entretanto, os riscos envolvidos são altíssimos.

O *software* livre permite reduzir as barreiras à entrada, diluindo custos e, ao mesmo tempo, estimulando a sua adoção por usuários insatisfeitos com as práticas monopolísticas presentes no mercado, refletidas em relações custo/benefício por eles consideradas não vantajosas. Um bom exemplo é o Linux, que vem permitindo que uma série de empresas distribuidoras penetrem no mercado de *software* para estações de usuários, até aqui dominado quase exclusivamente pela Microsoft, com mais de 90% do mercado. De acordo com as práticas tradicionais da indústria de *software*, somente uma empresa do porte da Microsoft poderia aventurar-se a fazê-lo, uma vez que o custo estimado do desenvolvimento de um sistema baseado em Linux – calculado segundo padrões vigentes nos Estados Unidos e na Europa – é de US\$ 2 bilhões. Vale observar que cerca de 70% do custo de desenvolvimento de um produto de *software* são devidos a gastos com mão-de-obra, não estando nele computados outros custos característicos dos pacotes, referentes a *marketing*, divulgação, rede de distribuição etc., que, como visto, podem chegar a mais de dois terços do custo de um pacote de *software*.

Adoção do Software Livre por Empresas

São inegáveis para os usuários as vantagens associadas aos menores custos das licenças do *software* livre. Entretanto, ele não será capaz de atender a todas as necessidades de *software*. As empresas continuarão a fazer uso de *software* proprietário quando este for estratégico para o seu tipo de negócio ou lhe conferir uma vantagem competitiva. Alguns *softwares* de segurança continuarão também a ser proprietários pela necessidade de preservar o segredo do seu funcionamento.

De forma geral, a adoção do *software* livre por uma empresa traz os seguintes benefícios:

- melhor preço para os serviços associados à distribuição do produto, pois o acesso ao código-fonte por várias empresas concorrentes acaba tendo um efeito regulador sobre os preços desses serviços;
- garantia de continuidade do produto, pois não há o risco de a empresa proprietária do *software*, por qualquer razão, retirar-se do mercado;
- independência de fornecedor único, uma vez que, por ser aberto e livre, várias empresas podem distribuir o *software* e realizar os serviços de modificação necessários;
- melhor aproveitamento do *hardware* existente, por não existirem pressões para atualização de versões, inexoravelmente associadas a limitações ao suporte disponível para versões antigas; no caso do *software* livre, pelo fato de o código-fonte ser de acesso livre, sempre haverá uma empresa disposta a efetuar os serviços de manutenção e suporte;
- elevada qualidade do *software*, verificada nos casos em que existe um grande número de colaboradores e empresas envolvidos com o projeto;
- maior facilidade de configuração e adaptação às necessidades da empresa frente aos pacotes proprietários, que são desenvolvidos visando a um uso padrão e incorporam um grande número de funções sofisticadas, das quais cada usuário individual usará apenas uma pequena parcela, variável entre os grupos de usuários; e
- maior segurança do produto de *software*, pois, pelo fato de o seu código-fonte ser aberto, examinado por um grande número de pessoas, é praticamente nula a possibilidade de existência de *back doors*, brechas que permitem a sua invasão não autorizada.

É bem conhecida a relutância dos usuários em relação a trocas de *software*, decorrente não apenas da necessidade de treinamento, o que inclui quedas de produtividade própria e dos colegas consultados, bem como da necessidade de incorporação do legado

(dados ou documentos existentes) ao novo *software*, o que pode requerer, em alguma medida, retrabalho.

Assim, no caso de migração de *software* proprietário para *software* livre, como em qualquer caso de migração de *software*, uma empresa deve computar não apenas os custos da licença e dos serviços de consultoria, instalação e suporte associados, como também os seguintes fatores:

- necessidades de *hardware*, aquisição e serviços;
- necessidade e disponibilidade de aplicativos;
- equipe técnica; e
- custos invisíveis, que normalmente ocorrem uma única vez, associados a quedas de produtividade por necessidade de treinamento e desenvolvimento de hábitos, adaptação do legado e possível instabilidade dos novos sistemas.

Quanto aos serviços associados ao *software* livre, é importante observar que pode não ser interessante a manutenção na empresa de uma equipe técnica destinada ao estudo dos códigos-fonte, instalação, implementação de modificações, atualização de versões e treinamento. Essas são tarefas geralmente contratadas a empresas especializadas em serviços, das quais uma das atribuições é também a elaboração de documentação para os usuários.

O Software Livre e o Governo

Na maioria dos países em desenvolvimento o governo é o grande usuário individual de TI, o que torna a sua posição frente ao *software* livre decisiva para o sucesso desse movimento nos respectivos países. Por exemplo, de acordo com representante da Novell, empresa participante de projetos de *software* livre, 25% do faturamento da empresa no Brasil provêm de órgãos ligados ao setor público. Entretanto, a importância do governo deriva também de outras duas vertentes, já que a ele corresponde um triplo papel como desenvolvedor de *software*, grande demandante e formulador de políticas.

Muitas das aplicações de que o governo necessita são de uso naturalmente restrito, por serem específicas de determinadas instâncias do poder público. Cabe, assim, ao governo desenvolvê-las, seja diretamente, seja terceirizando parcial ou totalmente os projetos. A utilização de *software* livre nos projetos específicos de governo é aconselhável sempre que possível, pela adoção de padrões abertos, o que implica um menor custo, bem como o atendimento da responsabilidade de manutenção de dados sobre os cidadãos, os quais não podem ficar à mercê de formatos proprietários. De outra parte, não

é aceitável que o governo exija dos cidadãos a apresentação de dados em formato proprietário, independentemente de qual ele seja.

Nos casos em que o setor público aparece como demandante, devido aos orçamentos limitados, o *software* livre significa uma solução de baixo custo, com expressiva economia nos gastos com licenças. São válidos também todos os benefícios apresentados para o uso do *software* livre em empresas. Além disso, a grande flexibilidade de uso e cópia proporcionada pelo *software* livre tornam-no extremamente atraente em um ambiente de compras controladas como é o setor público.

Verifica-se no setor público uma redução do custo de migração do *software* proprietário para o livre, em particular nas atividades de ensino. Não existe por parte dos alunos a fixação em padrões previamente aprendidos, o que torna a sua adaptação ao *software* livre muito mais imediata. Isso significa, ao longo do tempo, um importante reforço ao processo de penetração do *software* livre no mercado tradicional de *software*.

Por outro lado, a priorização do apoio do governo às iniciativas de *software* livre deve ser adequadamente ponderada pelos diversos projetos possíveis. Se são inegáveis as vantagens da adoção do *software* livre em várias aplicações, isso não pode impor restrições ao crescimento da indústria de produtos de *software* no país, a qual está baseada no tradicional licenciamento proprietário.

A concentração da indústria de *software* nos Estados Unidos configura nos outros países, principalmente nos periféricos, uma situação de grande dependência. Isso confere à adoção do *software* livre por esses países um caráter estratégico pelo seu poder de fomentar o desenvolvimento tecnológico e o emprego qualificado em produtos de *software* e serviços.

A formação de recursos humanos é uma grande beneficiária do *software* livre, o qual permite que os códigos-fonte de produtos em desenvolvimento, majoritariamente sujeitos, como visto, a métodos de trabalho e testes profissionais, sejam estudados e alterados por alunos dos cursos de TI. O contato direto com o trabalho desenvolvido pelas comunidades certamente contribui para que esses cursos de formação sejam confrontados com padrões de qualidade mundial.

O fato de as comunidades desenvolvedoras serem integradas por colaboradores situados em sua maioria nos Estados Unidos e na Europa faz com que as soluções adotadas sejam de cunho mais geral, distantes das necessidades brasileiras. A construção de produtos voltados para as peculiaridades brasileiras ou sul-americanas requer a criação de projetos de *software* livre nacionais e regionais.

Por outro lado, a análise das potencialidades do *software* livre faz surgir a questão da sua utilização como instrumento estratégico no esforço de erradicação do fosso digital que divide ricos e pobres, estes cada vez mais à margem das novas tecnologias da informação.

A esse respeito, é importante mencionar o lançamento, ainda neste ano de 2004, do Programa Brasileiro de Inclusão Digital (PBID), já apresentado à Câmara de Política de Desenvolvimento Econômico, o qual tem por objetivo o acesso dos cidadãos das classes C, D e E às tecnologias da informação. O PBID deverá contemplar investimentos em telecentros, geridos pelas próprias comunidades, disponibilizando à população computadores conectados à internet. O uso de *software* livre nesses telecentros é também um requisito, como forma de torná-los economicamente viáveis. Junto aos telecentros haverá oficinas de reparos de computadores e espaços de produção audiovisual multimídia. O PBID planeja implantar mil telecentros até junho de 2005, estando prevista no Plano Plurianual de Investimentos (PPA) a instalação de seis mil dessas unidades até 2007.

Conclusão

O setor de *software* tem passado por profundas transformações desde a sua separação do *hardware*, fato que marcou o nascimento de uma indústria independente. Desde então um grande dinamismo tem se manifestado, levando o *software* a outros setores e aplicações não imaginados.

Duas forças antagônicas manifestam-se na base desse dinamismo. A primeira está presente no movimento de expansão do *software* por inovações e usos pioneiros, executando funções ainda não automatizadas, o qual cria oportunidades para a entrada em cena de novas empresas. A segunda apresenta-se no crescimento da utilização do *software* em aplicações consagradas, consolidando empresas e padrões preexistentes.

Uma das características do *software* é apresentar o efeito de rede que, tal como ocorre, por exemplo, na telefonia ou na internet, faz com que o valor de uma rede aumente mais que proporcionalmente ao número dos seus usuários. Isso implica que um determinado padrão de *software* esteja tão mais consolidado quanto maior for o número dos seus usuários, transformando-se, pelo efeito de rede, em um fabuloso obstáculo ao estabelecimento de outros padrões, o que é especialmente importante no caso de o padrão dominante ser proprietário.

A disseminação de um padrão de *software* requer um determinado número de usuários para que possa se estabelecer o efeito de rede. Sabedoras desse fato, as novas empresas investem pesadamente em *marketing*, buscando convencer os possíveis

usuários das vantagens da mudança, e na criação de uma economia de rede, tais como serviços de distribuição, suporte e treinamento. Uma rede estabelecida possui fortes sinergias entre produtos e serviços associados, configurando um ecossistema que, a um tempo, indica e reforça a existência da rede.

Assim, o lançamento de produtos que operam dentro de um padrão estabelecido, por um lado, beneficia-se da economia e, por outro, reforça o padrão. Já produtos que requerem a mudança de padrão têm que enfrentar uma forte resistência, relativa a custos do usuário – decorrentes de novas licenças, necessidade de treinamento, queda de produtividade, tratamento de dados e sistemas legados etc. – e potencializada pela economia de rede.

A adesão de um grande número de usuários a um *software* proporciona efeitos de economia de escala muito importantes por permitir a diluição dos custos de desenvolvimento do produto, *marketing*, comercialização e outros, ao mesmo tempo em que propicia margens necessárias ao constante reinvestimento, característico da indústria de *software*. Tal reinvestimento tem por objeto: P&D de atuais e novos produtos; atualização de infra-estrutura (plataformas de *hardware* e *software*) e ferramentas; capacitação de recursos humanos; etc.

A existência de uma marca conhecida e de uma reputação associada à qualidade e à continuidade da empresa é perseguida tanto por empresas de produto quanto de serviços. Da mesma maneira que a redução de custos de gerenciamento e desenvolvimento por aprendizagem com a experiência, que eleva as margens das empresas.

Em suma, a necessidade de escalas mínimas, a necessidade de capital para investimento em tecnologia e despesas com *marketing* e comercialização, a diferenciação de produtos e serviços, a criação de uma rede de serviços associados, entre outras, constituem barreiras à entrada normalmente encontradas em outros setores e que existem também no de *software*. Tais barreiras são, porém, potencializadas pela economia de rede, o que é tanto mais sensível quanto mais padronizado é o segmento em que ela se estabelece. De forma análoga, segmentos mais customizados costumam ser menos sensíveis à economia de rede. Como exemplos extremos de um e outro caso, podem ser citados o formato de arquivo de planilha *x/s* e um aplicativo sob encomenda cujo mercado restringe-se a um único usuário.

A penetração do *software* livre altera significativamente esse quadro por basear-se em padrões abertos, o que faz com que a economia de rede não mais constitua uma barreira à entrada de novas empresas no mercado. Por outro lado, é menor a necessidade de capital para investimentos em P&D, o que faz diminuir a escala

mínima requerida. A diferenciação perde sentido em grande parte, podendo surgir uma competição por preços.

Quanto ao custo de mudança, a migração de um *software* proprietário para um *software* livre costuma possuir um custo menor que uma migração entre *softwares* proprietários distintos por não envolver aquisição de licenças. Entretanto, nos casos em que houver uma grande similaridade de uso entre o novo *software* e o antigo, o custo de migração poderá ser bastante reduzido, como é o caso na substituição do Unix pelo Linux.

A jovem e pequena indústria de *software* nacional apresenta uma forte feição de provedor de serviços. Isso em boa parte decorre do fato de ela ter surgido num momento em que a maior parte do mercado brasileiro já estava ocupada pelas corporações internacionais, cujos produtos traziam escalas imbatíveis de seus países de origem. Os serviços, por seu turno, precisam da proximidade entre provedor e usuário, o que obriga a realização dessa atividade no país.

A indústria nacional possui bons e variados produtos aplicativos, contudo voltados para segmentos específicos – verticais – ou que demandam fortes componentes de consultoria e serviços, como os ERPs de pequeno porte. Como visto, não bastam boas idéias, pois as barreiras à entrada no mercado de *software* são muitas. Assim, a maioria desses produtos ou é fortemente ligada a serviços ou é destinada a mercados de empresas menores.

Também não devem ser esquecidas as perdas advindas de desenvolvimentos interrompidos ou que não conseguiram atender adequadamente aos requisitos especificados no projeto do produto. Foi visando minimizar a ocorrência desses tipos de problemas que foram criados modelos de avaliação do processo de desenvolvimento, cuja certificação constitui hoje uma forte barreira à entrada no segmento de serviços de *software*, especialmente no mercado internacional.

O surgimento do *software* livre abre novas perspectivas para a indústria nacional, uma vez que traz consigo paradigmas de desenvolvimento diferentes do *software* proprietário, propiciando oportunidades a novas empresas em segmentos já ocupados. Representa também a oportunidade de criar, a baixo custo, aplicações para o atendimento de demandas sociais, especialmente aquelas voltadas para a erradicação da exclusão digital.

De maneira geral, a adoção do *software* livre pode ser incentivada por ações de governo, na medida em que este preconize a sua opção em compras públicas. Cabe observar que os governos possuem elevada participação na demanda do setor em países em desenvolvimento.

É necessário também que o governo tenha bem claro, no seu papel de formulador de políticas, quais são os espaços públicos e quais são os instrumentos de fomento à atividade privada, no sentido do fortalecimento da indústria nacional de produtos de *software*, não apenas de serviços.

Novas oportunidades estão continuamente surgindo, como produtos em segmentos ainda não automatizados, a criação de produtos inovadores ou o fato de a mão-de-obra brasileira ser mais barata que a dos países centrais. Entretanto, é necessária a profissionalização da indústria nacional, refletida em processos modernos e eficientes de administração, controle e produtividade para que as oportunidades possam ser aproveitadas de forma duradoura.

Referências Bibliográficas

- ACCENTURE. *Business process outsourcing big bang – creating value in an expanding universe*. Aug. 2002 (disponível em www.accenture.com).
- BABAIE, E. *Worldwide IT services market definitions guide, 2004*. Gartner Dataquest Guide, May 2004 (disponível em www.gartner.com).
- BATES, M. D., DAVIS, K. B., HAYNES, D. D. Reinventing IT services. *The McKinsey Quarterly*, n. 2, p. 143-153, 2003.
- BUSINESS WEEK. The Infotech 100. June 21, 2004.
- BUSINESS WEEK ON-LINE. How Microsoft can embrace Linux. July 26, 2004 (<http://www.businessweek.com>).
- CAMPBELL-KELLY, M. *From airline reservations to Sonic the Hedgehog – a history of the software industry*. Massachusetts Institute of Technology, 2003.
- CHAKRAVORTI, B. The new rules for bringing innovations to market. *Harvard Business Review*, p. 59-67, Mar. 2004.
- CRUZ, G. *Will your industry embrace or reject offshore services?* Gartner, Oct. 2003 (disponível em www.gartner.com).
- EGAN, T. *Structural change and spatial dynamics of the U.S. software industry*. ICF Consulting – Information Technology Publications, Apr. 1998 (disponível em www.icfconsulting.com).
- FAGUNDES, J., PONDÉ, J. L. *Barreiras à entrada e defesa da concorrência: notas introdutórias*. Rio de Janeiro: Universidade Federal do Rio de Janeiro, s/d.
- FRICK, S., NUNES, R. Produtos, estruturas de mercado e estratégias competitivas no setor de *software*. *Economia e Empresa*, São Paulo, v. 3, n. 1, p. 34-44, jan./mar. 1996.

- GARTNER. *Software market research methodology and definitions, 2003-2004*. Gartner Dataquest Guide, Jan. 2004 (disponível em www.gartner.com).
- _____. *2004 press releases* (vários) (disponível em www.gartner.com).
- GAZETA MERCANTIL. Novell aposta no crescimento do Linux. São Paulo, 2 de setembro de 2003.
- GORNICK, L. *Introdução ilustrada à computação*. 1985. [Tradução do *The cartoon guide to computer science*.]
- GRAHAM, C., LATIMER, N., BISCOTTI, F., CORREIA, J., ESCHINGER, C., PANG, C., TOPOLINSKI, T. *Software market research methodology and definitions, 2003-2004*. Gartner Dataquest Guide, Jan. 16, 2004 (<http://www.gartner.com>).
- GUERRA, A. C., ALVES, A. M. *Aquisição de produtos e serviços de software*. Rio de Janeiro: Elsevier, 2004.
- HEXSEL, R. A. *Software livre: propostas de ações de governo para incentivar o uso de software livre*. Universidade Federal do Paraná, 2002 (Relatório Técnico).
- IDC. *40 years of IT – looking back, looking ahead*. 2004 (disponível em www.idc.com).
- INOVAÇÃO. Empresa vai à justiça cobrar licença por trechos do GNU-Linux. Unicamp, 6 de julho de 2004 (<http://www.inovacao.unicamp.br>).
- INTERNATIONAL INSTITUTE OF INFONOMICS. *Free/libre open source software: survey and study*. The Netherlands: University of Maastrich, July 2002 (Final Report) (<http://www.infonomics.nl>).
- JORNAL DO COMERCIO. Grandes conglomerados utilizam o software gratuito. São Paulo, 3 de junho de 2004.
- KLEPPER, R., JONES, W. O. *Outsourcing information technology, systems & services*. 1998.
- LINDER, J., JACOBSON, A., BREITFELDER, M. D., ARNOLD, M. *Business transformation outsourcing: partnering for radical change*. Accenture/Institute for Strategic Change, July 2001 (disponível em www.accenture.com/isc).
- MACCORMACK, A. Product-development practices that work: how internet companies build software. *Mit Sloan Management Review*, Cambridge, Winter 2001.
- MASSACHUSETTS INSTITUTE OF TECHNOLOGY. *A indústria de software no Brasil 2002: fortalecendo a economia do conhecimento*. Coordenação Geral Brasil: Sociedade Softex Campinas, 2002.

- MCCOY, D., CALDWELL, F., COMPORT, J., DRESNER, H., CORREIA, J., ANDREWS, W., SMITH, D., CALDWELL, B., GRIGG, J., PRING, B., ROUSSEL, A., CANTARA, M. *Where to invest and what to avoid in software and services*. Gartner, Feb. 2004 (disponível em www.gartner.com).
- NAMBISAN, S. Why service business are not product business. *Mit Sloan Management Review*, p. 72-80, Summer 2001.
- O ESTADO DE S.PAULO. Sistema Linux já é coisa de gente grande. São Paulo, 13 de abril de 2004.
- PALM BEACH POST. Linux to benefit from law in China. Mar. 14, 2004 (<http://www.palmbeachpost.com>).
- RAMALHO, J. A. *Introdução à informática: teoria e prática*. São Paulo: Futura, 2003.
- ROSELINO, J. E. *Uma análise das potencialidades da atividade de software no Brasil à luz das práticas concorrenciais no setor*. Campinas, 1998 (Dissertação de Mestrado).
- ROSELINO, J. E., GOMES, R. O *software* e as cadeias produtivas internacionalizadas. In: Furtado, J. (org.). *Globalização das cadeias produtivas do Brasil*. São Carlos: EdUFSCar, 2003.
- SILICON VALLEY BIZ INK. *Apache HTTP server reaches record eight consecutive years of technical leadership*. July 29, 2004 (<http://www.svbizink.com>).
- STEINMUELLER, W. E. *The U.S. software industry: an analysis and interpretative history*. Mar. 1995.
- TAURION, C. C. *Software livre: potencialidades e modelos de negócio*. Rio de Janeiro: Ed. Brasport, 2004.
- U.S. DEPARTMENT OF COMMERCE. *Digital economy 2003 – economics and statistics administration*. Dec. 2003.
- VERZELLO, R., REUTER III, J. *Processamento de dados*. São Paulo: McGraw-Hill, 1984, vol. II.
- YOUNG, A. *Management update: outsourcing market view, what the future holds*. Gartner, June 2004 (disponível em www.gartner.com).

Sites Consultados

<http://encyclopedia.thefreedictionary.com>

<http://httpd.apache.org>

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

<http://www.distrowatch.com>
<http://www.freebsd.org>
<http://www.fsf.org>
<http://www.gnu.org>
<http://www.ibm.com>
<http://www.isc.org>
<http://www.itautech.com.br>
<http://www.linex.org>
<http://www.linux.org>
<http://www.microsoft.com>
<http://www.netbsd.org>
<http://www.nsf-middleware.org/Middleware>
<http://www.openbsd.org>
<http://www.openoffice.org>
<http://www.opensource.org>
<http://www.osdl.org>
<http://www.sei.cmu.edu>
<http://www.softwarelivre.org>
<http://www.sun.com>