

# Editorial

Caro leitor,

Há poucos anos testemunhamos no país o surgimento da discussão sobre o chamado *software* de código aberto, também conhecido como *software* livre.

O debate, inicialmente restrito aos meios universitários, rapidamente se expandiu e ganhou agenda obrigatória em todos os fóruns especializados em tecnologia da informação.

Hoje, o que assistimos é a uma discussão com uma pauta extensa. Entre os temas que atraem a atenção podemos citar: a definição de padrões de desenvolvimento compartilhado de programas e sistemas com código aberto, as regras de comercialização e uso dos códigos produzidos e, além disso, os seus inevitáveis desdobramentos em propostas de políticas públicas de informática. As questões de segurança, custos de implantação e manutenção, interoperabilidade, garantia de evolução dos produtos e direitos dos usuários, entre outros, são aspectos que despertam a atenção e são também objeto de estudos e deliberações.

No entanto, o tema decisivo que arde no centro da discussão é o embate entre duas visões aparentemente conflitantes sobre o tratamento que se deve dar ao fruto do trabalho intelectual, no caso, programas de computador. Uma delas acolhe como premissa o direito indiscutível de propriedade do autor sobre sua obra, com todas as implicações daí resultantes. A outra, em vertente oposta, advoga o direito social ao conhecimento, ampliando as possibilidades de uso do produto intelectual através de outorga de direitos.

Por essa razão e de forma previsível, já que a natureza do assunto induz a isso, o debate de imediato transcendeu as fronteiras da discussão técnica e alçou de forma fulminante a esfera do embate político.

A conseqüência disso, que tem sido a marca registrada desse debate, é a forma quase apaixonada com que a discussão tem sido travada.

A revista Fonte não poderia se ausentar do debate. Ao contrário, entra nele animada pela relevância do tema e seguindo as premissas de isenção e aprofundamento cuidadoso no assunto. A abordagem procurou registrar as diferentes posições políticas, ideológicas e técnicas que o assunto desperta.

Entre seus colaboradores, estão algumas das maiores autoridades no assunto e o resultado é a reprodução exata do que ocorre nos fóruns técnicos e políticos onde se dá a discussão.

Seguindo seus princípios editoriais, a revista Fonte abre espaço para todas as correntes, mas passa ao largo delas sem tomar partido. Essa tarefa cabe ao leitor.

Um abraço,

**Maurício Azeredo Dias Costa**

**FONTE**

Uma publicação da:



Volume 2 - nº 02 - Janeiro/Junho de 2005



**Governador do Estado de Minas Gerais**  
Aécio Neves da Cunha  
**Secretário de Estado de Planejamento e Gestão**  
Antonio Augusto Junho Anastasia  
**Diretor-Presidente**  
Maurício Azeredo Dias Costa  
**Diretora de Projetos e Negócios**  
Glória Maria Menezes Mendes Ferreira  
**Diretor de Tecnologia e Produção**  
Raul Monteiro de Barros Fulgêncio  
**Diretor Administrativo e Financeiro**  
José Ronaldo Fidelis  
**Diretor de Desenvolvimento Empresarial**  
Nathan Lerman

#### CONSELHO EDITORIAL

Antonio Augusto Junho Anastasia  
Maurício Azeredo Dias Costa  
Marcio Luiz Bunte de Carvalho  
Amílcar Vianna Martins Filho  
Gustavo da Gama Torres  
Paulo Kléber Duarte Pereira  
Marcos Brafman

#### EDIÇÃO EXECUTIVA

**Assessoria de Comunicação**  
Isabela Moreira de Abreu  
**Assessoria da Presidência**  
Pedro Marcos Fonte Boa Bueno  
**Coordenação do Projeto Editorial, Gráfico e Publicitário**  
Gustavo Grossi de Lacerda  
**Edição, Reportagem e Redação**  
Isabela Moreira de Abreu - MG 02378 JP  
**Colaboração**  
Luís Carlos Silva Eiras  
**Universidade Corporativa Prodemge**  
Enilton Ferreira Rocha  
Marta Beatriz Brandão P. e Albuquerque  
Luiz Cláudio Silva Caldas  
**Consultoria Técnica**  
Sérgio Melo Daher  
**Revisão**  
Graça Kind  
**Projeto Gráfico e Diagramação**  
Carlos Weyne  
**Consultoria Gráfica**  
Guydo José Rossi Cardoso de Meneses  
**Capa**  
Guydo José Rossi Cardoso de Meneses  
Gustavo Grossi de Lacerda  
**Fotolito e Impressão**  
Policrom / Gráfica Formato  
**Tiragem**  
Três mil exemplares  
**Periodicidade**  
Semestral

PATROCÍNIO



A revista Fonte visa à abertura de espaço para a divulgação técnica, a reflexão e a promoção do debate acerca de visões plurais no âmbito da tecnologia da informação, sendo que o conteúdo dos artigos publicados nesta edição são de responsabilidade exclusiva de seus autores.

Prodemge - Rua da Bahia, 2277 - Bairro Lourdes  
CEP 30160-012 - Belo Horizonte - MG - Brasil

[www.prodemge.mg.gov.br](http://www.prodemge.mg.gov.br)  
[prodemge@prodemge.gov.br](mailto:prodemge@prodemge.gov.br)



Planejamento e Gestão

# Diálogo

## *Propriedade intelectual e democratização do conhecimento: as novas perspectivas do direito autoral*



Ronaldo Lemos, mestre em Direito pela Universidade de Harvard, doutor em Direito pela USP, diretor do Centro de Tecnologia e Sociedade da Escola de Direito da Fundação Getúlio Vargas e diretor do projeto Creative Commons no Brasil. Em Harvard, trabalhou no Berkman Center for Internet & Society. É autor de vários artigos publicados no Brasil e no exterior e de dois livros, "Comércio Eletrônico" (2001) e "Conflitos sobre Nomes de Domínio e Outras Questões Jurídicas da Internet" (2003), ambos publicados pela Editora Revista dos Tribunais. Lançará este ano seu terceiro livro, "Direito, Cultura e Tecnologia".

**A** polêmica questão jurídica dos direitos autorais aplicados à indústria do *software* suscitou uma série de outras questões e ampliou a discussão para outras formas de criação e difusão da produção cultural, dando origem ao movimento internacional Creative Commons. Nesta edição, o Diálogo é com o professor Ronaldo Lemos, referência nacional em direito na internet e diretor do projeto Creative Commons no Brasil.

Ele fala do projeto, que teve sua origem na Universidade de Stanford, com o professor Lawrence Lessig, hoje presente em 25 países. Além de imprimir uma nova dinâmica às relações entre autores e consumidores, o movimento estabelece um novo modelo de negócios, baseado na liberdade do autor de estipular quais direitos ele está disposto a conceder sobre sua obra, flexibilizando o conceito do *copyright* - de "todos os direitos reservados" para "alguns direitos reservados".

Nesta entrevista, Ronaldo Lemos fala da tendência mundial da lei do direito autoral de restringir o acesso ao conhecimento, através do conceito de propriedade intelectual, e de como setores da sociedade, preocupados com as barreiras impostas à disseminação da produção cultural e científica, se mobilizam para reverter essa situação, mostrando o balanço do Creative Commons no mundo, no licenciamento de textos, imagens, filmes, músicas, livros e outras obras.

Ronaldo Lemos defende a produção de *software* livre pelo poder público como política pública governamental, fundamentada pela Constituição Federal e pela política nacional de informática, e explica como o conteúdo da produção intelectual pode ser compartilhado, preservando-se os direitos do autor.

**Fonte:** *Na sua opinião, qual a base filosófica, conceitual, para a definição de propriedade e democratização do conhecimento?*

O conceito-chave é propiciar e manter os incentivos adequados para a criação de novas obras intelectuais, ao mesmo tempo em que se propicia um amplo acesso para o conhecimento. Isso não é tarefa fácil. No panorama atual, a idéia de propriedade intelectual está se tornando absoluta. Mais absoluta até, em alguns casos, do que a propriedade de um bem físico.

Isso traz distorções que prejudicam o acesso ao conhecimento, especialmente na medida em que caminhamos para um mundo cada vez mais digital.

Note que propriedade é um conceito jurídico histórico, forjado com relação a bens materiais, do qual o direito moderno se apropriou e passou a aplicar para os bens intelectuais. A mera importação desse conceito já traz distorções, quanto maior for o aprofundamento do mesmo. Eu confio muito na capacidade de imaginação institucional, que possa permitir criar um conceito próprio para lidar com os bens intelectuais de maneira apropriada, ponderando incentivos para a criação e direito de acesso.

**Fonte:** *Como conciliar a democratização dos conteúdos sem ferir os direitos do autor?*

Um dos modos como isso pode ser feito é através das licenças Creative Commons, um projeto criado pelo professor Lawrence Lessig, da Universidade de Stanford, hoje presente em mais de 25 países do mundo. A idéia é justamente essa: manter os direitos autorais do autor e criador da obra intelectual e ao mesmo tempo permitir à sociedade como um todo

ter alguns direitos sobre a obra. Nesse sentido, ao adotar uma licença Creative Commons, o autor opta por um modelo de "alguns direitos reservados", que é diferente do tradicional

"todos os direitos reservados". Por esse último, a sociedade não possui nenhum direito de acesso. Pelo modelo do Creative Commons, a sociedade possuirá os direitos de acesso que o dono da obra especificar.

**Fonte:** *Cite aspectos específicos das licenças de software.*

As licenças de *software* livre inspiraram a criação do Creative Commons. O *software* livre foi criado quando o autor de

---

A idéia é justamente esta:  
manter os direitos autorais do autor e criador da obra intelectual e ao mesmo tempo permitir à sociedade como um todo ter alguns direitos sobre a obra.

---

um *software* decidiu conferir à sociedade direitos de modificação, acesso ao código-fonte, cópia e distribuição. Ao contrário disso se tornar um ato filantrópico, surgiu um modelo de negócios totalmente consistente. Só a IBM, por exemplo, já investiu mais de 2 bilhões de dólares no *software* livre. Empresas como a Sun e a Novell estão indo pelo mesmo caminho. Foi a partir do exemplo bem-sucedido do *software* livre que o projeto Creative Commons foi criado. Apesar das licenças do projeto serem muito mais variadas, dando muito mais opções para o autor quanto aos direitos que ele pretende conceder, o conceito original é comum.

**Fonte:** *Com essa nova dinâmica proposta para a produção de software, como ficam as relações comerciais: quais seriam as distinções entre os modelos de negócios na produção e comercialização de sistemas livre e proprietário?*

Como mencionei, o Creative Commons e o *software* livre demonstram claramente a existência de um novo tipo de mercado. O Creative Commons pode potencializar o valor econômico de obras culturais. Tome por exemplo a cidade de

---

Na área de textos, professores da FGV estão licenciando livros pelo Creative Commons. Tais licenças permitem, por exemplo, que a cópia do livro seja autorizada.

---

Olinda, onde existe um grande projeto de disponibilizar o conteúdo cultural produzido por lá na internet, tudo sob o Creative Commons. Olinda quer se tornar a primeira cidade Creative Commons do Brasil. Uma das iniciativas é documentar o patrimônio histórico da cidade, como, por exemplo, o oratório do Mosteiro de São Bento, e licenciar livremente as fotos, que poderão ser amplamente circuladas. O Mosteiro de São Bento foi a peça central da exposição do Museu Guggenheim de

Nova Iorque na exposição Brazil Body & Soul. Há milhares de pessoas no mundo interessadas em saber como está o oratório e obter material que possa ser usado do mesmo. Esse é só

um dos exemplos: conjugado com a força que Olinda tem em artes plásticas, música, teatro e festas públicas, como o carnaval, você tem um conteúdo riquíssimo, pronto para ser visto e aproveitado mundialmente. E o Creative Commons é ferramenta fundamental para facilitar a disseminação, utilização e transformação livre desse conteúdo. Dessa forma, o modelo de licenciamento Creative Commons pode, na verdade, potencializar os aspectos econômicos de várias obras intelectuais.

**Fonte:** *Qual o balanço do Creative Commons no Brasil e no mundo? Há estimativas de emissão de licenças? Que percentual refere-se a software?*

A estimativa é que no mundo hoje existem mais de 6 milhões de obras licenciadas pelo Creative Commons. Essas obras incluem texto, imagens, filmes, músicas, livros e outras. No Brasil já existem inúmeras obras, muitas delas de extrema relevância. Por exemplo, documentários, como o filme "Casa de Cachorro", vencedor do festival "É tudo ver-

dade", é licenciado em Creative Commons. Bandas como o "Gerador Zero", ou o "Mombojó", de Recife, também usam a licença, para não se mencionarem

as músicas do ministro Gilberto Gil. Na área de textos, professores da FGV estão licenciando livros pelo Creative Commons. Tais licenças permitem, por exemplo, que a cópia do livro seja autorizada, resolvendo a questão com relação a obras copiadas em xerox.

**Fonte:** *Comparando com outros países, como a cultura brasileira assimila esses novos conceitos?*

O Brasil é um dos países mais importantes no cenário do Creative Commons,

mas não é o único. As licenças possuem um impacto enorme nos Estados Unidos, onde estão sendo incorporadas em *websites* que lidam com conteúdo produzido descentralizadamente, como o Flickr e o GarageBand. O primeiro é como o fotolog, só que com maior alcance nos EUA. O segundo é um *site* para que bandas disponibilizem seus materiais *on-line*. Ambos disponibilizam uma interface para que as obras postadas já sejam desde o início licenciadas em Creative

Commons. Como se não bastasse, o Yahoo agora tem um dispositivo de busca especializado em buscar obras em Creative Commons, que pode ser acessado pelo endereço

<http://search.yahoo.com/cc>. Isso tudo para não mencionar o licenciamento das obras de gente como Marshall Sahlins, o mais importante antropólogo no mundo, em Creative Commons, dentre outros.

**Fonte:** *A lei pode mesmo facilitar as mudanças paradigmáticas necessárias para a evolução do movimento pelo conhecimento livre? Se não, qual seria o papel do CC?*

O Creative Commons surge de uma constatação pessimista: a lei, a partir da

década de 90, tem sido sistematicamente modificada para restringir o acesso ao conhecimento. Isso significa que a idéia de propriedade tem sido aplicada de forma cada vez mais absoluta. Além disso, estão sendo criados "metadireitos autorais", que são estruturas jurídicas que se aplicam mesmo a obras em domínio público, restringindo o acesso até mesmo a elas, como mostra o exemplo do chamado *broadcast flag* e da diretiva sobre direitos *sui generis* aplicáveis a bancos de dados na Europa. Com a lei funcionando como forma de restrição, resta a iniciativas como o Creative Commons gerar modelos mais balanceados, mostrando que esses modelos são viáveis. Esse balanço ocorre graças à mobilização voluntária de criadores de obras intelectuais.

**Fonte:** *As licenças específicas, como a GPL, figuram no direito oficial, são legais? Explique como é, na prática, o trabalho do Creative Commons.*

Tanto a GPL quanto o Creative Commons são totalmente legais. Isso porque ambos se fundam em um princípio muito básico, de que o autor, enquanto dono de todos os direitos de uma criação

intelectual, tem por consequência a faculdade de fazer o que quiser com esses direitos. Dessa forma, tanto o Creative Commons quanto o *software* livre fundamentam-se no modelo tradicional do direito autoral. Tudo o que os autores fazem é autorizar alguns direitos sobre as obras, direitos esses que pertencem por lei ao próprio criador da obra.

**Fonte:** *A internet exige a criação de uma legislação específica? Como é o entendimento jurídico do mundo virtual? Em que medida o direito tem acompanhado a dinâmica de evolução da internet?*

A internet tem mostrado que precisa não só de uma, mas de várias leis específicas. Leis essas que, em primeiro lugar, garantam os direitos de acesso à informação, protegendo-os de abusos e pressões por parte da indústria do conteúdo. Essa é, inclusive, a posição de professores como Gunther Teubner, um dos maiores juristas da atualidade, que defende a constitucionalização do direito ao acesso e à informação, retirando-o da arena das brigas políticas em torno da propriedade intelectual. No entanto, essa idéia ainda está longe de acontecer.

**Fonte:** *Pode-se dizer que o movimento do software livre estimulou a criação do Creative Commons?*

Sim, como mencionei, o Creative Commons possui a mesma fonte do *software* livre. Apesar disso, as licenças do Creative Commons são mais variadas do que aquelas usualmente empregadas no licenciamento do *software* livre. Isso, porque o mercado de bens culturais possui peculiaridades muito próprias. A vedação, por exemplo, do uso comercial da obra, que é uma das características possíveis das licenças do Creative Commons, não existe no *software* livre. E o curioso é notar que essa cláusula tem justamente um objetivo antimonopolístico, de consolidar redes de comunicação sem fins comerciais, como as redes *peer-to-peer* e a tecnologia *bit torrent*. Por isso, essas cláusulas são fundamentais quando se fala em cultura.

**Fonte:** *O senhor lançará um novo livro sobre o assunto. Fale um pouco sobre a abordagem de "Direito, Cultura e Tecnologia".*

Meu livro "Direito, Cultura e Tecnologia" será lançado este ano pela editora FGV. Naturalmente, o livro será licenciado em Creative Commons, de modo

que qualquer pessoa esteja autorizada a fazer cópias e circular seu conteúdo. O livro trata de questões como modelos alternativos à propriedade intelectual tradicional, o *software* livre, o *Creative Commons*, o papel dos provedores da internet, modelos de direitos autorais que o Brasil não deveria seguir, dentre outros assuntos.

**Fonte:** *Qual a sua opinião sobre a polêmica questão da participação do Estado na construção de software livre?*

A questão do Estado na construção do *software* livre não é polêmica, ela é uma política pública governamental, fundamentada pela Constituição Federal e pela política nacional de informática, definida em lei. O fomento por parte do Estado ao *software* livre conjuga-se com a proibição estabelecida pela lei de informática da criação de situações monopolísticas. O Estado, ao incentivar uma rede descentralizada de desenvolvimento de *software*, promovendo o acesso e disseminação do conhecimento técnico, está atendendo a essa política. Dessa forma, incentivar e manter o pluralismo de modelos econômicos no desenvolvimento de *software* faz parte das atribuições constitucionais do Estado.

---

Tudo o que os autores fazem é autorizar alguns direitos sobre as obras, direitos esses que pertencem por lei ao próprio criador da obra.

---

# Software Livre

## (Free Software)

O termo **Software Livre** se refere aos softwares que são fornecidos com as liberdades de executá-los, estudá-los, modificá-los e redistribuí-los (com ou sem alterações).

Entre polêmicas de natureza tecnológica, política, econômica e ideológica, ampliadas para questões relacionadas à propriedade do conhecimento, o *software* de código aberto – livre ou não – vem trilhando em todo o mundo um caminho de crescimento, especialmente nos últimos dez anos. As discussões sobre essa forma diferente de desenvolvimento e distribuição de *software*, consolidada pelo uso da internet, ganha as mais diversas dimensões, diretamente influenciadas pelos diferentes interesses e posicionamentos em pauta.

As previsões para o *software* de código aberto vão desde a perspectiva de um modismo, passando pelo conceito de tendência tecnológica e diversificação natural da indústria de *software*, até a idéia de um novo modelo de desenvolvimento que se perpetuará, deixando para o *software* proprietário, hoje detentor de 95% do mercado, a posição de solução cara para poucos usuários.

O modelo traz consigo novos termos e a multiplicação de instituições representativas dos diversos setores envolvidos ou atingidos direta ou indiretamente pelo *software* de código aberto; consolida uma nova forma de desenvolvimento, que valoriza a colaboração, e propõe uma discussão que extrapola os limites do *software*, enfatizando, em todo o mundo, a problemática da propriedade da produção intelectual e do conhecimento.

O desenvolvimento colaborativo está, no entanto, longe de ser uma novidade na história da humanidade. Como afirma o gerente do Centro de Computação da Unicamp, professor Rubens Queiroz, "Toda a nossa civilização foi construída através do compartilhamento de informações". A internet, no entanto, transformou de forma radical a dinâmica dos relacionamentos.

O teórico das novas mídias, professor no Departamento de Artes Visuais da Universidade de San Diego, Lev Manovich, em estudo sobre a questão da autoria, lembra que a internet traz uma variedade de novos modelos, com formas diversas de colaboração, mas afirma que "A autoria colaborativa é, numa perspectiva histórica, uma norma e não uma exceção", referindo-se especificamente à construção de catedrais e à produção de filmes. Para ele, novas formas de autoria, novas relações entre produtores e consumidores e novos modelos de distribuição da produção cultural são consequências naturais das novas indústrias e culturas de mídia.

A consolidação de uma dinâmica inovadora de desenvolvimento e distribuição de *software* tem suscitado, porém, polêmicas acirradas e posições contundentes. Elas são justificadas por seus adeptos como falta de informação ou como reação normal das grandes empresas que adotam o modelo proprietário, com alto grau de dependência da comercialização de licenças, sua maior fonte de receitas. Segundo essa visão, o *software* livre é uma ameaça à estabilidade dos negócios dessas empresas.

Alguns defensores do modelo proprietário, por sua vez, caracterizam o desenvolvimento aberto como um ambiente de caos, onde a eventual falta de um responsável formal pelos produtos distribuídos deixaria seus usuários desamparados e os sistemas vulneráveis, pagando por esse desserviço valores mais altos que aqueles requeridos pelo *software* proprietário.

O movimento natural do mercado de *software* em todo o mundo, no entanto, alicerçado numa

visão de negócio, com foco no desenvolvimento de novos produtos que atendam a um mercado emergente, ultrapassa as barreiras ideológicas e apaixonadas e se adapta a esse novo cenário. Ao lado de empresas que se ocupam exclusivamente do desenvolvimento de produtos livres, como a SugarCRM, TrollTech, MySQL e SleepyCat, grandes empresas, como a HP, IBM, Novell, Oracle e Consist, tradicionalmente desenvolvedoras do modelo proprietário, abrem seus mercados, ampliam portfólios e criam departamentos específicos para atender à demanda crescente por soluções abertas, sejam elas livres ou não.

Embora não sinalize uma adaptação a essa nova modalidade, o próprio vice-presidente mundial da Microsoft, Bradford Smith, em entrevista ao jornal Estado de São Paulo (21/02/05), conclui que "Os dois modelos estão mais parecidos a cada ano que passa: os *softwares* de código aberto se tornam cada vez mais comerciais e os *softwares* comerciais, mais abertos", afirmou.

Da mesma forma, o presidente da Microsoft do Brasil, Emilio Umeoka, em entrevista à publicação Executivo de Valor (abril 2005) prevê que o *software* comercial e o *software* livre caminham para uma junção. "Há movimentos como o da (americana) Novell, que comprou a (alemã) SUSE e o da (francesa) Mandrakesoft, que adquiriu a (brasileira) Conectiva. Os jogadores tornam-se multinacionais e o modelo de comercialização fica mais tradicional. O cliente tem de comprar licenças, atualizações e assim por diante. Por outro lado, o *software* comercial está aprendendo a trabalhar em comunidade. A Universidade Federal de Pernambuco e a Unesp, de São Paulo, desenvolvem códigos para a Microsoft, como resultado de uma licitação da qual participaram 17 universidades. Também promovemos o acesso ao código-fonte. No caso das urnas eletrônicas, demos acesso ao governo, partidos de oposição e situação. É cada vez mais transparente quais serão nossos principais competidores."

Divulgação



Luiz Fernando Maluf, da Sun

Para o gerente de Novas Tecnologias Aplicadas da IBM Brasil, Cezar Taurion, o movimento do *software* livre vai transformar a indústria de tecnologia da informação, mas não de forma homogênea. Segundo ele, ao afetar a estrutura de grandes empresas, esse novo modelo obriga a revisão de práticas comerciais. Ele defende a idéia de que as empresas que atuam intensamente em serviços podem rapidamente aumentar sua receita entrando fortemente nesse novo segmento, diversificando sua atuação e deixando para os usuários a tarefa da escolha.

Há ainda as empresas que já nasceram com a visão do desenvolvimento colaborativo, como a Sun Microsystems, que se inspirou, desde a sua criação, em 1981, na idéia do desenvolvimento compartilhado como ferramenta de qualidade e agilidade na criação de soluções. O diretor de

Desenvolvimento Estratégico da Sun no Brasil, Luiz Fernando Maluf, garante que o conceito de desenvolvimento compartilhado cria mercado e dá opções aos usuários, enquanto o *software* proprietário cria produtos e conceitos de *marketshare* e monopólio. "Cria mercado na medida em que você tem a convivência de vários fornecedores de um mesmo produto; tem a opção, por exemplo, de adquirir o Java de um universo enorme de fornecedores diferentes. Você pode escolher."

O compartilhamento para criar novos produtos em rede traz ainda o benefício do encurtamento do ciclo de desenvolvimento, fazendo com que novas soluções cheguem mais rapidamente ao mercado, explica Luiz Fernando Maluf. "O ciclo tecnológico que era de aproximadamente três anos para desenvolver e disponibilizar um novo produto, foi reduzido para quatro meses com o desenvolvimento compartilhado; uma empresa trabalhando em padrões normais não consegue essa produtividade. Temos portanto a economia de rede, que obtém resultados que uma empresa sozinha não conseguiria."

Ainda a favor do *software* de código aberto, Maluf lembra que ele agrega uma importância estratégica ao negócio, ao permitir a adaptação do *software* às necessidades específicas do usuário, e também pelo fato de permitir entender a sua lógica, desenvolver programas e evitar o monopólio.

O desenvolvimento compartilhado é justamente a premissa da tecnologia Java, desenvolvida pela Sun, baseada na computação em rede. Em artigo nesta edição sobre a história do Java, que comemora dez anos de mercado, o especialista da Sun Microsystems, Michael Yuan, lembra que liberar o código gratuitamente "É uma das maneiras mais rápidas de gerar solução em larga escala e também uma ótima maneira de obter a ajuda da comunidade de desenvolvedores para inspecionar código e encontrar *bugs*".

## Software Livre e Open Source x Software Proprietário

Isabela Abreu



Thadeu Cascardo, do Movimento Minas Livre

Considerada por alguns adeptos do *software* livre um artifício do chamado FUD - Fear, Uncertainty and Doubt (medo, incerteza e dúvida) - estratégia utilizada para desestimular o entendimento e adoção do modelo aberto, a confusão em torno das denominações dadas a esse novo modelo de desenvolvimento e distribuição de *software* é uma realidade.

A falta de entendimento em torno dos termos esconde, na verdade, a filosofia que orienta duas linhas semelhantes, mas distintas, de posicionamento: o *software* livre e o código aberto (ou *open source*).

A história começa com Richard Stallman, o projeto GNU e a Free Software Foundation, concebida no início da década de 80, no Massachusetts Institute of Technology – MIT, nos Estados Unidos. O termo *free*, ao ser traduzido para o português, trouxe a dualidade de significados da palavra em inglês: livre e gratuito, originando um entendimento distorcido do seu real sentido. Na concepção de seu criador e de seus seguidores, *software* livre "é matéria de liberdade, e não de preço".

Segundo um dos mais atuantes desenvolvedores do Movimento Minas Livre, Thadeu Cascardo, no caso do *software* livre há uma definição ideológica de defesa da liberdade do conhecimento. "A criação da Free Software Foundation, em 1984, já trouxe parâmetros claros que definem o *software* livre e as quatro liberdades que o caracterizam: liberdade de uso (pessoal, acadêmico ou comercial), de distribuição (que permite fazer cópias e distribuí-las), de estudo ou modificação (adequar o *software* para necessidades específicas) e a liberdade de distribuir essas modificações. Isso tem, portanto, relação com o direito autoral, que por princípio impede todas essas coisas."

Já o Código Aberto, ou *Open Source*, é uma iniciativa de 1998, da Open Source Initiative – [www.osi.org](http://www.osi.org) – no sentido de se encontrar um termo mais adequado e tentar torná-lo mais comercial.

Mas diferentemente do *software* livre, que traduz uma ideologia de liberdade, o *open source* reflete uma maior preocupação com a qualidade na produção de *software* e com sua segurança, na medida em que mais pessoas podem verificar o código. O enquadramento nessa categoria pressupõe a definição de dez critérios (veja quadro).

Para Thadeu Cascardo, são definições distintas em sua essência, "Mas se você pegar as quatro liberdades do *software* livre e os dez critérios do *open source*, vai ver que eles têm tanta coisa em comum que a distinção está, na verdade, na filosofia que sustenta cada uma das tendências."

Exemplificando: muitas vezes o *software* está disponível para quem quiser usá-lo, mas não permite que o usuário agregue suas idéias e sugestões. Nesse caso, ele é *open source*, mas não é livre. "Na minha opinião, as duas coisas têm que andar juntas. Apesar de permitir um desenvolvimento mais colaborativo, em que mais pessoas podem interagir, há divergências e também a necessidade de convergências", explica Cascardo.

Em oposição a essas duas tendências está o chamado *software* proprietário, que funciona com a venda de licenças de uso do produto da forma como foi adquirido, ou seja, não é permitido ao usuário ter acesso ao código-fonte. Esse é o modelo presente na grande maioria das estações em todo o

mundo, liderado pela norte-americana Microsoft. A disseminação de seus produtos em escala global, principalmente o Windows e o Office, coloca a empresa na mira dos defensores do *software* de código aberto, com o rótulo de monopólio, fato que é refutado pelo vice-presidente mundial da Microsoft, Bradford Smith, em entrevista ao Estado de São Paulo (21/02/2005), na qual ele afirma que "Nós somos uma grande empresa e nossos produtos são populares."

Thadeu Cascardo considera o termo *software* proprietário inadequado: "Isso implica que há uma propriedade intelectual e na minha opinião o conhecimento não deveria se enquadrar como propriedade. No direito romano – explica – havia o conceito de privado, público e comum. Comum não é necessariamente público; nenhuma instituição seria considerada representativa do conhecimento. A produção intelectual deveria se enquadrar como propriedade comum, ela não é privada nem pública."

## As quatro liberdades e os dez critérios

### As quatro liberdades do *software* livre

1. A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)
2. A liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades (liberdade nº 1). Acesso ao código-fonte é um pré-requisito para essa liberdade.
3. A liberdade de redistribuir cópias de modo que você possa ajudar o seu próximo (liberdade nº 2).
4. A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

(Fonte: *Wikipedia*)

### Os dez critérios que definem um *software open source*

*Código aberto não significa apenas acesso ao código-fonte. Os termos de distribuição do software de código aberto devem estar de acordo com os seguintes critérios:*

#### 1. Redistribuição Livre

A licença não deve restringir qualquer das partes de vender ou integrar o *software* como componente de uma distribuição de *software* agregada, contendo programas oriundos de diversas fontes. A licença não deve exigir *royalties* ou qualquer outro tipo de custo para venda.

#### 2. Código-Fonte

O programa deve incluir o código-fonte e precisa permitir a distribuição tanto na forma de código-fonte como na compilada. Quando alguma forma do produto não é distribuída com o código-fonte, precisa dispor de meios reconhecidos de obtenção do código por não mais que um preço de custo razoável para a reprodução, ou *download* pela internet sem custos. O código-fonte deve ser a forma privilegiada na qual um programador altera o programa. Código-fonte deliberadamente obscurecido não é aceito. Formas intermediárias, como a saída de um processador ou tradutor, não são permitidas.

#### 3. Trabalhos Derivados

A licença deve permitir modificações e trabalhos derivados e precisa garantir a estes a distribuição sob os mesmos termos como a licença do *software* original.

**4. Integridade do Autor do Código-Fonte**

A licença deve restringir a distribuição do código-fonte sob forma alterada somente se permitir a distribuição de *patch files* com código para o propósito de modificar o programa quando de sua compilação. A licença deve permitir explicitamente a distribuição do *software* compilado a partir de um código modificado. A licença pode exigir que trabalhos derivados tragam um nome ou versão distintos da original.

**5. Sem Discriminações Quanto a Pessoas ou Grupos**

A licença não deve discriminar qualquer pessoa ou grupo de pessoas.

**6. Sem Discriminação Quanto a Ramos de Empreendimento**

A licença não deve restringir nenhum uso a ramos de empreendimento específicos. Por exemplo, ele não pode ser vedado para uso comercial ou para uso em pesquisas genéticas.

**7. Distribuição da Licença**

Os direitos associados ao programa devem ser aplicáveis a todos aos quais o programa é redistribuído sem a necessidade de licenças adicionais a estas partes para sua execução.

**8. A Licença Não Deve Ser Específica ao Produto**

Os direitos associados ao programa não devem depender deste ser parte de uma distribuição particular de *software*. Caso o programa seja extraído desta distribuição e usado ou distribuído através dos termos de sua licença, todas as partes para as quais o programa é redistribuído devem gozar dos mesmos direitos, garantidos na conjunção original da distribuição do *software*.

**9. A Licença Não Deve Restringir Outros Softwares**

A licença não deve aplicar restrições sob outros *softwares* que sejam distribuídos com *softwares* objetos de seu licenciamento. Por exemplo, a licença não deve insistir que outros programas distribuídos no mesmo meio sejam *software* de código aberto.

**10. A Licença Deve Ser Neutra às Tecnologias**

Nenhuma aplicação da licença deve preferir uma tecnologia específica ou estilo de interface.

(Fonte: *Open Source Initiative*)

**Origens:** Bruce Perens escreveu o primeiro rascunho desse documento como "As Diretrizes do Software Livre Debian", e o aprofundou a partir dos comentários dos desenvolvedores do Debian em uma lista de discussão em junho de 1997. Ele removeu as referências específicas ao Debian nesse documento e criou a "Definição de Código Aberto".

## Afinal, free é gratuito?

A gratuidade, por exemplo, não é requisito para nenhuma das duas tendências: não figura expressamente nas quatro liberdades do *software* livre nem nos dez critérios do *open source*.

O professor Roberto Hexler, do Departamento de Informática da Universidade Federal do Paraná, em texto publicado no *site* do Governo Federal, explica que

"É importante não confundir *software* livre com *software* grátis porque a liberdade associada ao *software* livre de copiar, modificar e redistribuir independe de gratuidade. Existem programas que podem ser obtidos gratuitamente mas que não podem ser modificados, nem redistribuídos. Por outro lado, existe a possibilidade de uso não-gratuito em todas as categorias."

## A questão das licenças

---

Especialistas afirmam que a questão do *software* livre é, na verdade, de natureza jurídica, e não tecnológica. Isso porque no momento da aquisição de um modelo livre ou proprietário, o foco vai estar na forma de licenciamento adotada. A lei de direito autoral garante ao autor de qualquer produção intelectual todos os direitos sobre sua criação. Isso quer dizer que a lei impede, por princípio, a cópia, distribuição ou modificação de qualquer obra científica ou artística – música, livro ou *software*.

A novidade introduzida no licenciamento de *softwares* de código aberto é justamente a iniciativa do autor de ceder, a quem adquire seu produto, alguns direitos além dos básicos garantidos pelo direito autoral – que se limitam ao uso de um *software*, leitura de um livro ou à audição de uma música, sem no entanto poder reproduzi-los ou copiá-los.

As licenças de *software* são, na verdade, contratos legais em que autor e usuário explicitam as condições de uso de determinado produto. "Não é 'casa de mãe Joana'" – explica o diretor da EAC Software, empresa que representa a MySQL no Brasil, Helvécio Borges Filho. "Muitas pessoas usam o *software* livre sem conhecê-lo muito bem; pensam que as coisas funcionam sem regras, sem critérios. Não é assim. Há inclusive o fenômeno da pirataria do *software* livre: apesar da existência de licenças, de uma ordenação jurídica, muitas vezes o produto é utilizado de forma irregular, numa clara contravenção às regras do licenciamento."

A classificação de um *software* como livre ou *open source* está vinculada, portanto, à avaliação não do *software*, mas da licença sob a qual ele é liberado. "Você só pode fazer algumas

coisas a partir do momento em que tem a licença prévia e expressa do autor", ressalta Thadeu Cascardo. Quando o autor licencia o *software*, o que está fazendo é justamente dar essa autorização prévia e expressa, sob condições que ele define.

Analisando a questão cultural no Brasil com relação à compreensão e uso das licenças, o professor Ulisses Leitão, diretor de Tecnologia do Instituto Doctum ([www.doctum.com.br](http://www.doctum.com.br)), considera que "Não há como não enxergar que a reação do brasileiro leigo ao paradigma do *software* livre é limitada pela mentalidade da chamada Lei de Gerson. Assim, a primeira reação é de espanto, e logo vêm conclusões precipitadas de que alguém vai enganar alguém, fazendo com que o Linux deixe de ser livre, ou mesmo que a Microsoft, pelo seu poderio econômico, poderia tomar o código do Linux e fechá-lo". Para o professor Ulisses, esse pensamento demonstra um desconhecimento dos mecanismos jurídicos da licença GPL, que garante a impossibilidade de que esses dois fatos ocorram.

Ele avalia também a questão da pirataria, considerada "enorme" no Brasil. "Muitas vezes a discussão é se a pessoa faria a opção por um produto livre em detrimento de um produto proprietário pirateado. Assim a pirataria dá a ilusão de que os produtos são baratos. Acho que se não houvesse a pirataria, se as pessoas tivessem que realmente pagar cada centavo das licenças que usam, a consolidação e mesmo hegemonia do *software* livre no Brasil seriam muito mais significativas, porque a relação de preços praticados no País, versus poder de compra da população, inviabilizaria a utilização do *software* proprietário."

## *Licenças copyright e copyleft*

---

As licenças impõem, portanto, algumas condições. Uma delas é a chamada *copyleft* – uma brincadeira com a palavra *copyright* – direito de cópia. O professor Roberto Hexler explica (softwarelivre.gov.br) que a maioria das licenças usadas na publicação de *software* livre permite que os programas sejam modificados e redistribuídos. "Essas práticas são geralmente proibidas pela legislação internacional de *copyright*, que tenta justamente impedir que alterações e cópias sejam efetuadas sem a autorização do autor. As licenças que acompanham *software* livre fazem uso da legislação de *copyright* para impedir utilização não-autorizada, mas essas licenças definem clara e explicitamente as condições sob as quais cópias, modificações e redistribuições podem ser efetuadas, para garantir as liberdades de modificar e redistribuir o *software* assim licenciado. A essa versão de *copyright* dá-se o nome de *copyleft*."

Para Helvécio Borges Filho, o Brasil é lento no processo de entendimento das licenças. Um complicador dessa situação, segundo ele, é a forma pouco atraente desses documentos: "A GPL - General Public License, por exemplo, a licença mais adotada, é indigesta, seu entendimento jurídico é complicado". Há cerca de 80 outras licenças, mas a GPL é a mais utilizada. Luiz Fernando Maluf,

da Sun, explica que há licenças como a CDDL - Common Development and Distribution License, em que o cliente obtém o código-fonte mas com a condição de não redistribuí-lo, a fim de manter a padronização do produto.

Dentro da idéia de que a licença é uma autorização expressa para uso de um *software*, há casos em que um mesmo produto é comercializado sob licenças diferentes. No caso de produtos desenvolvidos por empresas, se as determinações da GPL forem satisfatórias para a necessidade de uso do interessado, ele tem o *software* livre e gratuito. Caso exista alguma peculiaridade não atendida pela GPL, a empresa pode disponibilizar o produto com outra licença que contemple essa necessidade, oferecendo uma "licença não GPL" pela qual o interessado vai pagar. Trata-se, portanto, de um produto com duas licenças diferentes, ou licenciamento dual.

A finlandesa MySQL, fornecedora do banco de dados aberto mais utilizado no mundo, faturou em 2004 US\$15 milhões, sendo 65% da receita da empresa originários da venda de licenças não GPL, informa Borges. São feitos cerca de 40 mil *downloads*/dia. Entre os usuários do banco de dados estão quase que exclusivamente clientes da iniciativa privada, principalmente o setor de telecom.

## *Como ganhar dinheiro com software livre?*

---

Isso responde uma das perguntas freqüentes sobre o *software* livre: de onde vem o dinheiro? Uma das chaves dessa questão está no fato de que a venda de licenças de *software* livre pode ser a maior fonte de renda de uma empresa do setor. Há outras

empresas no mundo que adotam esse mesmo modelo híbrido de negócio. São dois pilares de sustentação: um na licença GPL e outro na não GPL.

Vale dizer que o mercado de *software* livre também é movimentado por dinheiro. Outra fonte

de receita é o espaço para doações de patrocinadores das comunidades existentes na internet: há nos *sites* que abrigam projetos de desenvolvimento compartilhado *links* para doação. Na medida em que uma pessoa ou empresa economiza, por exemplo, mil dólares na compra de um *software*, ela doa parte

dessa diferença, que é reinvestida para o desenvolvimento, com a compra de novos equipamentos. "O cenário social em que muitos desenvolvedores estão não é o brasileiro", constata Helvécio Borges. "São, em sua maioria, pessoas que não estão ligadas ao dinheiro, que trabalham voluntariamente."

## *A dinâmica do desenvolvimento colaborativo*

---

Em todo o mundo, um contingente atuante e crescente de pessoas dedica-se ao desenvolvimento do *software* livre, identificando eventuais *bugs*, propondo melhorias e novas implementações. A internet contribuiu de forma decisiva para a consolidação dessa forma de desenvolvimento, em que várias pessoas podem contribuir e sugerir. "Trata-se de um tipo de comunidade que forma um movimento bastante interessante, pessoas que estão provavelmente no topo da Pirâmide de Maslow, em busca de sua auto-realização profissional", conclui Helvécio Borges.

Há basicamente dois grupos responsáveis pela produção do *software* de código aberto, estruturados num modelo que vale para todo o mundo: desenvolvedores de projetos da "comunidade", gerenciados por pessoas físicas interessadas, mas que adotam uma estrutura hierárquica semelhante à de qualquer organização; e projetos desenvolvidos por empresas. Em ambos, o produto final pressupõe o código-fonte aberto, independentemente do tipo de licença. Entre 70 e 80% desses projetos são regidos pela licença GPL. Quando um *software* é desenvolvido dentro da modalidade livre, ele deve estar obrigatoriamente cadastrado num *site* gerenciador de projetos. Existem *sites* específicos para essa finalidade e há sempre um autor, um gerente do projeto. No Brasil, uma das principais referências é o [www.codigolivre.org.br](http://www.codigolivre.org.br).

O gestor do projeto define o tipo de licença – GPL, por exemplo – que vai regular determinado produto. "Há regras, como num contrato de aluguel", argumenta Helvécio Borges. Para facilitar o entendimento, ele sugere a comparação desses *sites* de desenvolvimento com um "grande balaio, com códigos-fonte disponíveis para qualquer pessoa, desde que ela acate as regras da licença que define o produto de seu interesse."

Vale lembrar que um brasileiro, Marcelo Tossati, é atualmente o mantenedor da versão 2.4 do *kernel* do Linux. Ele foi indicado pelo próprio criador do Linux, Linus Torvalds, para cuidar da parte mais nevrálgica e importante de qualquer sistema operacional, o *kernel*. Paranaense, de Curitiba, Marcelo Tossati é o responsável pela avaliação e validação de sugestões relativas à versão do Linux, recebidas de membros da comunidade de todo o mundo.

No caso dos projetos gerenciados por empresas, há equipes de desenvolvedores de *software* que estão em estruturas organizacionais convencionais; são profissionais contratados, que recebem salários, em organizações como a IBM e a MySQL. Há ainda empresas que trabalham com equipes próprias e com a comunidade da rede, como a Sun, por exemplo. Segundo Luiz Fernando Maluf, a empresa participa da JCP – Java Community Process ([www.jcp.org](http://www.jcp.org)), consórcio que reúne 500

empresas do porte da Intel, IBM, Siemens e Oracle para o desenvolvimento em Java sob uma mesma especificação, o que impede a formação de um mercado monopolizado. Esses padrões são especificados por grupos abertos à participação de empresas, instituições ou indivíduos. A comunidade JCP reúne no mundo cerca de 4,5 milhões de desenvolvedores, sendo, desse total, entre 70 e 100 mil no Brasil.

A Fundação JBoss ([www.jboss.org](http://www.jboss.org)) é outro exemplo de associação de organizações de grande porte para financiamento de projetos de código aberto, sendo detentora de contratos fabulosos. "Há negócios ligados a *software* livre plenamente sustentáveis, rentáveis, com aplicações tanto para uso comercial quanto não comercial", diz Helvécio Borges. "Há empresas que financiam projetos, como a IBM, que possui um centro de excelência em Linux. Ela adequou-se ao mercado e diversificou sua linha de produtos. No caso da adoção do Linux, devido a seu código aberto, ela pode adequar suas soluções. Não é algo movido a emoção, mas a racionalidade."

O diretor-presidente da Softex, Márcio Girão Barroso, ao comentar a pesquisa sobre *software* livre realizada pela entidade, em parceria com a Unicamp e Ministério da Ciência e Tecnologia, afirma que uma das demonstrações de que o modelo do *software* livre não é uma utopia é a sua viabilidade econômica. "Porque algo utópico não tem

sustentação. A pesquisa realizada mostra que o *software* livre realmente nasceu de uma maneira utópica, mas tomou força em função de pessoas que estavam insatisfeitas com a hegemonia de alguns produtos no mercado. Podemos dizer que estamos agora numa terceira fase do desenvolvimento do *software* livre, em que é possível fazer bons negócios: você pode alavancar negócios, construir empresas com o desenvolvimento."

Opinião semelhante tem o professor da Unicamp, Rubens Queiroz: "O que se vê é que é possível viver de *software* livre. A Solis é um exemplo eloqüente disso e sua experiência é hoje referência tanto no Brasil como no resto do mundo. Para entender um pouco mais sobre esse modelo de negócios vale a pena visitar o site da Solis – [www.solis.coop.br](http://www.solis.coop.br) – e ler o estudo de caso da Univates, que migrou totalmente para *software* livre", sugere Queiroz.

Em ambos os modelos – com origem na comunidade ou em empresas –, a regra é o desenvolvimento colaborativo, o controle de *bugs* e a realização de testes, a fim de garantir uma melhoria contínua dos *softwares*. Especialmente com relação aos produtos desenvolvidos pela comunidade, em caso de dúvida sobre qualquer determinação descrita na licença, Helvécio Borges aconselha: "Até que se esclareça, é melhor não usar."

## Fiscalização

---

Como então é feita a fiscalização de uso dos produtos de código aberto? A Free Software Foundation é referência a respeito da utilização de produtos livres, especialmente no que se refere a uma das quatro "liberdades" que definem o *software* livre, que é a de que o "bolo deve sempre crescer". A GPL é a licença menos permissiva, mas

a que possibilita maior perenidade. Ela garante que o usuário não se aproprie de algo sem retornar com um produto melhor; é a materialização da filosofia que fundamenta o *software* livre, amparada por um instrumento jurídico. A FSF intervém em todo o mundo, caso haja necessidade. Na verdade, o interesse maior não é na fiscalização, mas no incentivo

ao desenvolvimento colaborativo e na prevenção contra a apropriação indevida de produtos para comercialização como *software* proprietário.

Para o professor da Unicamp, Rubens

Queiroz, o modelo de *software* livre entende que o mercado é grande o suficiente para todos. "Não existe a pretensão de dominação do mundo com a obtenção de lucros indecentes", garante.

## Quando usar?

---

"*Software* Livre não é remédio para todos os males, mas para muitos. O Brasil ainda não tem essa maturidade e entendimento de como ele pode ser útil." A afirmação é do empresário Helvécio Borges. Para ele, o Brasil ainda está na fase de entendimento do licenciamento do produto. "Exploram-se pouco os recursos disponíveis. No exterior, o uso é intensivo", constata.

Analisando o mercado dos usuários, ele afirma que usa o *software* livre "quem tem mobilidade e entendimento objetivo de racionalização de recursos: em geral, empresas de médio e grande porte, as empresas privadas, em especial". Ele destaca no setor privado as empresas de telecom.

É o caso da finlandesa Nokia, que investe em *software* livre no Brasil. Segundo o presidente da organização, Fernando Terni, em entrevista à Revista Gallery Leaders (edição 4), a opção da empresa pelo *software* livre se fundamentou no reconhecimento internacional que o Brasil tem, hoje, como centro de desenvolvimento de *software* livre, principalmente no sistema operacional de código aberto GNU/Linux, e também pelo apoio que o governo Lula tem demonstrado à causa, "o que é muito importante. Por isso a nossa intenção e decisão de investir no Brasil. Por isso os finlandeses escolheram o país para torná-lo sede de mais um centro de desenvolvimento de aplicativos com base em *software* livre para celular (a disputa foi com a Índia, outro grande pólo mundial de desenvolvimento de *software*)".

Fernando Terni explica que o sistema dos aparelhos é proprietário da Nokia, mas com código aberto. "O que fazemos é fomentar a criação de *softwares* para rodar na nossa plataforma. E como se faz isso? Disponibilizamos recursos para que os desenvolvedores possam ter mais chances de sucesso. Então, temos não apenas plataformas de desenvolvimento, que disponibilizamos para serem usadas, como também damos treinamento a esses profissionais e ajudamos na parte da gestão do desenvolvimento de *software*", explica.

A experiência da EAC Software começou, na época de criação da empresa, com estações utilizando o Windows. "Quando houve necessidade de um *upgrade* – lembra o diretor da empresa – ficaria muito caro e teríamos que fazer um *upgrade* também do *hardware*, o que aconteceria de forma contínua. Optamos então por quebrar o ciclo do consumo; saímos dessa ciranda do império Wintel", ironiza Helvécio. Para ele, essa questão se agrava ainda mais quando "não queremos fazer *upgrade* do *software* e o fabricante não oferece suporte às versões antigas".

No setor público brasileiro ele aponta entre os maiores usuários de *software* livre as Forças Armadas, principalmente a Marinha. "São interessados, por questão de segurança. É nesse momento que o chamado 'xiita' está certo. A Marinha Brasileira, por exemplo, precisa ter o controle absoluto do tiro de um torpedo. No caso de um foguete que sobe ao espaço, da mesma forma, como ter segurança se o código não é conhecido? Há situações em que não

há opção, a indicação é usar o código aberto. É o caso também das urnas eletrônicas e do desenvolvimento nuclear, por exemplo: têm que estar no domínio absoluto do país."

Nas empresas públicas de tecnologia da informação, a adoção do *software* livre vem também conquistando adeptos. Exemplos de uso intenso nos âmbitos municipal, estadual e federal são mostrados nesta edição da Revista Fonte, em *cases* do Metrô-SP, da Prefeitura de Recife, Celepar (Paraná) e no Governo Federal. Segundo a presidente da Abep - Associação Brasileira de Entidades Estaduais de Tecnologia da Informação e Comunicação, Tereza Porto, "A adoção de *software* livre já é uma realidade dentro das empresas, em alguns casos" (Fatos Abep, abril 2005). Segundo ela, no Proderj - Centro de Tecnologia da Informação e Comunicação do RJ - a tecnologia já é adotada em soluções de infra-estrutura e segurança da informação e está em processo de implantação o Livre.RJ, como solução para automação de escritório.

Na opinião do professor Ulisses Leitão, é estratégico para o Estado ser o agente catalisador do desenvolvimento de *software* livre. "Há diversas razões para isso. Pela própria natureza do *software* livre, ele é uma propriedade da sociedade como um todo. Assim, a contratação de desenvolvimento de *software* pelo Estado, condicionando a empresa contratada a liberar o código e licenciá-lo como *software* livre, é a possibilidade de gerir uma política de atendimento às necessidades do país, garantindo o posterior desenvolvimento do mesmo, dentro da dinâmica do *software* livre."

Ele exemplifica, sugerindo um plano de produção de conteúdo e *software* educacionais livres, "extremamente estratégico para o país. O déficit educacional do país é imenso e é fundamental ampliar o acesso à educação em todos os níveis. Nesse caso, a pior saída - aliás, já tentada pelo MEC no finado ProInfo - seria a aquisição generalizada de *softwares* educacionais proprietários, atrelados a um sistema operacional proprietário. O fomento ao desenvolvimento de *softwares* educacionais livres,

baseados na *web* ou multiplataforma, traria enormes benefícios sociais e é, portanto, justificável como política pública".

A favor da capacitação em todas as tecnologias disponíveis, o diretor da EAC destaca também a questão de ampliação do poder de negociação que o domínio do código pode representar para uma empresa: "São os casos em que o sistema operacional aberto é estratégico; mesmo que você não vá desenvolver, o conhecimento do código aumenta o poder de negociação, o poder de barganha".

Lembra, no entanto, que "Quando há abundância de recursos, naturalmente as empresas não têm que analisar demais, deixam por exemplo o usuário escolher. Comprar uma bala a R\$ 0,05 ou a R\$ 0,10 aparentemente não faz muita diferença, mas trata-se de uma diferença de 100%. Há alternativas, e cabe ao gestor conhecê-las e avaliar o melhor uso, independentemente de ser livre ou proprietário. O Word Perfect, que é proprietário, por exemplo, custa R\$ 300. Ao adquirir o Word, o usuário paga R\$ 1.600 pelo pacote do Office, da Microsoft".

Ele explica que a experiência da gestão de TI em outros países mostra que o gestor tem realmente o poder de escolher a melhor tecnologia. "Mas em geral, no Brasil, quem dita as regras é o usuário, que na maioria das vezes sabe trabalhar com *software* de código proprietário, impõe sua vontade e não tem noção de quanto custa. Como cidadão e modesto conhecedor de TI, é duro ver um funcionário usando o pacote MS Office para escrever uma carta simples. Por que não usar o WordPad, que já faz parte do sistema operacional Windows? É essa irracionalidade e ingerência dos recursos que irrita aqueles que desfrutam de tudo que o mundo livre e proprietário oferecem. É uma situação difícil de reverter. Enfim, o ideal é a adoção de produtos diferentes para situações diferentes e esses produtos podem ser abertos ou não", conclui.

Para o gerente de Estratégias de Mercado da Microsoft, Roberto Prado, "Muitas vezes as escolhas levam em conta apenas o preço de aquisição do *software*, deixando de lado outros componentes que

entram na conta total de um orçamento de TI e que, na verdade, representam a maior parte dos gastos. É o que comprova um recente estudo realizado pela Bearing Point" (veja artigo de Roberto Prado nesta edição).

Já o professor Ulisses Azevedo Leitão destaca o aspecto do custo inicial: "Informações livremente compartilhadas, assim como o *software* livre, têm o seu custo de absorção nas empresas. É o custo

de consultoria, treinamento e capacitação. Assim, o mercado do *software* livre estará ligado intrinsecamente ao aumento da oferta educacional para capacitação, a formal e informal. Ou seja, Einstein nunca cobrou pelo conhecimento que ele gerou, mas quem quiser compreender a teoria da relatividade terá que adquirir livros e/ou cursar uma universidade. O mercado da educação é o grande mercado para a prestação de serviços em *software* livre".

Isabela Abreu



**Kássia Franco, da License**

Dirigindo a License Company, uma das empresas do grupo TBA – maior parceira Microsoft da América Latina e uma das maiores empresas de TI do país – Kássia Franco identifica uma série de vantagens nos produtos de conformação proprietária. Para ela, uma das questões importantes é justamente a continuidade do produto, sua evolução sustentada por um acompanhamento cuidadoso de uma mesma empresa, que reúne sugestões, críticas e o aperfeiçoa a cada versão. "Quando você vai comprar – explica Kássia Franco – tem a garantia de que a solução que você escolheu vai evoluir, a garantia de que tem um 'pai', um fabricante que garante a sua continuidade. A falta dessa paternidade gera problemas", argumenta.

## Os benefícios do proprietário

Roberto Prado, da Microsoft, concorda, afirmando que a facilidade de implementação, a compatibilidade, a confiabilidade e o suporte técnico da tecnologia escolhida são questões que fazem diferença, afetando diretamente o custo e o sucesso do projeto.

Atuando no setor há treze anos, a TBA também se adequa às necessidades do mercado e criou recentemente uma unidade de *software* livre. Segundo Kássia Franco, "Para acompanhar o mercado temos que estar sempre em movimento e já temos pedidos de produtos em código aberto". Com a experiência de representação de produtos como Oracle, Adobe e Citrix, que rodam em plataformas livres ou proprietárias, Kássia aprova o seu desempenho e segurança, especialmente em servidores.

Luiz Fernando Maluf, da Sun, é taxativo ao afirmar que, com relação ao aspecto de segurança, o Linux é muito mais evoluído e seguro. "Não há qualquer limite para criptografia, gerenciamento de acessos, controle de vírus e spam. O Linux é inerte a vírus. A questão da segurança está, na verdade, nas mãos do gerente de TI: se você tem um *software* utilizado por quatro mil usuários, eles terão acesso ao que o gerente de TI determinar. A arquitetura

do sistema operacional aberto é infinitamente superior".

Outro aspecto relacionado à segurança é apontado pelo diretor da EAC, que destaca a importância da auditabilidade, indisponível no *software* de código fechado. "A pseudo possível segurança existe por obscurantismo, por ocultamento do que foi implementado", afirma.

Kássia Franco aponta aspectos que considera fundamentais no momento da escolha entre as soluções aberta e proprietária, como por exemplo a facilidade de apoio para serviços continuados (ou suporte), desenvolvimento e infra-estrutura. Quanto

aos profissionais, ela acredita que o mercado do *mainstream* está melhor qualificado, por se tratar de uma cultura disseminada, o que faz com que a oferta de mão-de-obra seja mais satisfatória.

Com relação ao custo das duas soluções - aberta ou proprietária - Kássia Franco lembra que livre não é sinônimo de gratuito. "E mesmo quando é gratuito - argumenta - não podemos esquecer os custos com treinamento, suporte e desenvolvimento; isso pode sair mais caro por uma série de motivos." Para usuários de *desktop* ela argumenta que as soluções abertas exigem mais, uma vez que não apresentam interatividade.

## Saiba mais

Informações sobre os dois modelos de desenvolvimento - proprietário e livre - estão disponíveis nas mais diversas mídias, especialmente na internet. Além de explorar o aspecto comercial, alguns *sites* têm uma preocupação didática, expondo projetos, trabalhos acadêmicos e resultados de pesquisas relativas à sua área de atuação.

Confira as sugestões de endereço na seção Bookmarks (p. 41).



## Experiências na área pública em Minas

### Polícia Militar de Minas Gerais - Projeto Alferes Linux

Desde 2000 a Polícia Militar tem utilizado *software* livre em seus servidores de autenticação e de impressão em algumas de suas unidades. Segundo o chefe do Núcleo de Informática da corporação, Major Marcos Augusto Moreira e Silva, esse emprego baseava-se então na experiência pessoal de alguns de seus integrantes. Já em 2003 essa utilização tornou-se corporativa, com a migração do servidor de *e-mail* para o Linux.

"Em 2004 – lembra o Major Marcos – em razão das dificuldades orçamentárias do Estado para aquisição do *softwares* proprietários e/ou atualização dos já adquiridos, a Polícia Militar, através do Núcleo de Informática da Diretoria de Apoio Logístico, optou por uma solução baseada em *software* livre."

O projeto Alferes consiste no desenvolvimento de uma distribuição Linux baseada no produto Kurumim, composta por um conjunto de ferramentas de automação de escritório, navegação na internet e acesso aos sistemas corporativos, de fácil instalação e suporte.

"O desenvolvimento do projeto Alferes Linux está apenas começando, mas vislumbra-se, além da economia orçamentária, a capacidade de personalização, pois como se trata de *software* livre, o código-fonte está disponível para alterações", explica o gestor de Informática da PM.

Atualmente a distribuição já conta com vários aplicativos, entre eles o pacote de escritório, englobando editor de textos, planilha eletrônica e apresentação; *software* de navegação para a internet e *e-mail*; antivírus; emulador para o ambiente de

grande porte na Prodemge, e gerenciador de projetos. E está disponível em duas versões: uma para equipamentos antigos e com pouca memória, e uma outra para microcomputadores com maior capacidade de processamento e memória. "O que diferencia as duas distribuições é o ambiente gráfico que utilizam, sendo que na versão para microcomputadores mais antigos é utilizado o ambiente IceWM e nos demais, o KDE."

Hoje o Alferes Linux encontra-se instalado em praticamente todos os servidores das redes das unidades da Polícia Militar, além de um número maior em microcomputadores usados como estação de trabalho.

Com a experiência de cinco anos utilizando o *software* livre, o Major Marcos aconselha: "O Alferes Linux é um projeto grande, que merece a atenção de todos os órgãos do Estado, principalmente em razão da precariedade de obtenção de recursos orçamentários, com vista à manutenção da estrutura administrativa." Segundo ele, as empresas em geral têm procurado diminuir os seus custos, obtendo um maior lucro e mais recursos para investimentos. Dessa forma, o Estado, com a diminuição dos seus custos administrativos, teria condições de aplicar os seus recursos em áreas prioritárias.

Representantes de outras secretarias já visitaram a Polícia Militar com o objetivo de conhecer melhor o projeto e algumas delas utilizaram a distribuição para o desempenho de suas tarefas, "considerando que a distribuição Alferes Linux enquadra-se perfeitamente nas atividades que desenvolvem".



Major Marcos Augusto

Divulgação

## Secretaria de Estado de Educação

Na Secretaria de Estado de Educação, a implantação do *software* livre teve início em 2001, com a aquisição de 600 computadores com Linux, com recursos do Banco Mundial. Segundo o diretor de Recursos Tecnológicos da Secretaria, Paulo Mário Paiva Silveira, a principal motivação foi a redução de custos: "A renovação das licenças do Windows para todas as 42 regionais inviabilizaria continuar com o *software* proprietário."

A adoção do *software* livre passou num primeiro momento pelos aplicativos e, numa segunda fase, pelo sistema operacional, que já está quase que integralmente operando com Linux: mais de 90% dos servidores. A substituição do Windows pelo Linux vem sendo feita de forma gradual.

A implantação do novo modelo é apoiada por um programa de capacitação dos funcionários, tanto na capital quanto no interior. A Secretaria produziu em 2004 um CD que contém todo o conteúdo programático do treinamento, apostilas, dicas, documentos relativos à migração, que são distribuídos para as regionais na medida em que adotam a nova tecnologia. Atualmente, das 46 regionais, 23 já estão



Paulo Mário Silveira

Isabela Abreu

trabalhando com *software* livre. Foram capacitados 1035 funcionários. O objetivo é concluir as demais ainda em 2005.

Comemorando os bons resultados da experiência, o diretor de Recursos Tecnológicos da Secretaria, Paulo Mário, não descarta, no entanto, o uso do *software* proprietário: "Não penso em tirar o Windows", explica. "As tecnologias têm que caminhar juntas, procurando reduzir custos. Há situações em que o *software* proprietário é mais interessante, por exemplo nos contatos com o MEC: para o Censo Escolar, ele é necessário."

Por enquanto a filosofia é investir preferencialmente em capacitação, "E não em CDs", afirma, referindo-se às licenças. O projeto, que teve início contemplando a área administrativa da Secretaria já se estende também às escolas, onde é adotado para os laboratórios um modelo de um servidor com Linux atendendo terminais de trabalho (*thin client*) e oferecendo aos alunos da rede escolar acesso à internet e ferramentas do OpenOffice. O projeto prevê pelo menos 1600 escolas conectadas ainda este ano.

## Prodabel

Na Empresa de Informática e Informação do Município de Belo Horizonte – Prodabel – a experiência com *software* livre começou em 1995: o serviço de acesso à internet, correio eletrônico e provimento de conteúdo foram implantados utilizando-se o sistema operacional GNU/Linux. A partir daí várias outras soluções, como DNS e *cache* de páginas *web* foram implementadas, também usando *software* livre.

Em 2000 esse processo ganhou mais força com a criação de um grupo de incentivo ao *software* livre, definição de recomendações para seu uso e desenvolvimento de projetos mais abrangentes.

Seguindo as orientações desse grupo, a Prodabel decidiu evoluir o uso, que até então era apenas em servidores e começou o desenvolvimento de um *desktop* próprio, que fosse adequado ao uso da administração pública municipal. Segundo o

analista de Sistemas Márcio Bizzotto, "Toda a experiência com uso em servidores não era traduzida nos *desktops*, as distribuições disponíveis eram muito cheias de produtos desnecessários e de difícil instalação."

Em 2001 tiveram início experiências nos *desktops* com uso de *software* livre de uso genérico, com auxílio do Departamento da Ciência da Computação da UFMG, para a Secretaria Municipal de Educação, contemplando dois laboratórios (participando a SMED, Prodabel e DCC-UFMG) e duas escolas municipais (veja detalhes em [www.emcls.pbh.gov.br](http://www.emcls.pbh.gov.br)). Para Danilo Rodrigues César, analista de sistemas, "Aliando o conhecimento existente na Universidade à experiência de uso corporativo da Prodabel, tivemos um grande salto de qualidade e um ganho significativo de tempo."

Foi criado, então, o Libertas, em um projeto conjunto com a UFMG. Esse *desktop* foi implantado na área administrativa da empresa, experiência que resultou na criação de uma metodologia de



Márcio Bizzotto

migração, capacitação, apoio e suporte.

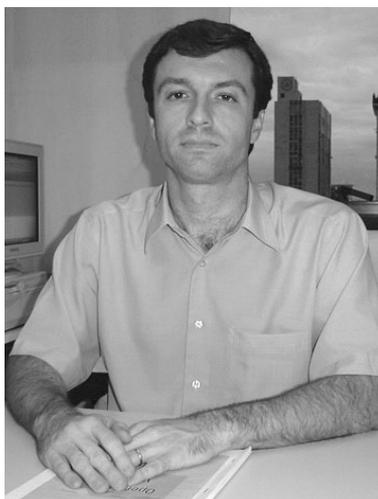
Para Márcio Bizzotto, um dos resultados mais significativos dessas iniciativas foi a informatização da rede municipal de Educação, toda baseada em *software* livre. São 182 escolas interligadas, contemplando áreas administrativas, bibliotecas e laboratórios pedagógicos. Um total de 190 mil alunos está recebendo uma identificação digital contendo *e-mail*, área de trabalho e *home-page*. Os dez

mil professores receberam capacitação específica em informática e têm hoje à sua disposição tecnologias de ponta, às quais muitos deles nunca tinham tido acesso. A Prodabel tem estrutura central para gerenciar as contas e os *e-mails* e monitorar todas as unidades. Tudo isso também implantado com *software* livre.

"O Libertas continua em evolução. Estão sendo agregadas funcionalidades que permitirão trabalhos pedagógicos com tecnologias digitais amplas, para produção de vídeos e rádio comunitária, dentre outras", conclui Bizzotto.

## Assembléia Legislativa

No Legislativo mineiro o processo de migração das máquinas do MS Office 97 para o OpenOffice.org começou em 2002 e atualmente cerca de 60% das estações já operam em *software* livre, inclusive os gabinetes de todos os deputados. A informação é do gerente geral de Sistemas de Informação da Assembléia Legislativa do Estado de Minas Gerais, Marcelo Migueletto de Andrade, que optou pelo uso do OpenOffice rodando com o Windows. Para



Marcelo Migueletto

recursos centrais, como servidores de bancos de dados e internet, é adotado o Linux e outros *softwares* livres.

A premissa que orienta o projeto na Assembléia Legislativa é a redução de custos com qualidade de serviços ao usuário. "Nossa maior preocupação – explica Migueletto – é não diminuir o nível de serviço e para isso estamos continuamente em busca de novas soluções. A nossa decisão é técnica e administrativa, portanto consideramos

as opções que melhor nos atendam, com a preocupação de reduzir custos. Estamos vendo o *software* livre com muito interesse e muita boa vontade."

O projeto contempla principalmente as estações de trabalho. "No *desktop* é que o *software* livre representaria maior economia", explica Marcelo Migueletto. "Além disso, a solução seria tecnicamente viável e apresentou a melhor relação custo/benefício. Para se ter uma idéia, uma cópia do Office custa quase 30% do valor de uma estação de trabalho; considerando cerca de 500 instalações de licenças que economizamos, são aproximadamente

R\$ 500 mil. Sem contar as atualizações anuais, que custam cerca de 29% do valor da licença."

A capacitação dos usuários foi também na Assembléia Legislativa um passo importante do processo: segundo Marcelo, constatou-se que os recursos do MS Office eram pouco utilizados. "O foco do treinamento no OpenOffice foi justamente nas funções que as pessoas realmente usavam." Houve ainda um trabalho de sensibilização para a mudança, mostrando os motivos e o planejamento. "A comunicação com os usuários foi útil para que pudéssemos contar com a vontade de contribuir."

## Interoperabilidade

---

Imagine o planeta Terra, com seus seis bilhões de habitantes falando idiomas diferentes mas se comunicando numa linguagem comum. Esse ideal no mundo do *software* se chama interoperabilidade: uma estrutura em que cada um é único e todos se entendem. A tecnologia moderna oferece condições para isso, garante o diretor da EAC, Helvécio Borges. "A proposta do Governo Federal deve ser principalmente interoperabilidade", explica. Isso porque, conforme o entendimento do empresário, o Governo não pode obrigar o uso de um padrão para que a sociedade acesse seus serviços: devem ser adotadas soluções que atendam a toda a população, sem discriminação.

"Um arquivo que não abre em qualquer estação é uma forma de aprisionamento. Isso não é bom, principalmente na área pública", defende Helvécio. "Ao divulgar um edital em formato eletrônico proprietário, o administrador estaria limitando o acesso àqueles que possuem o programa que abre aquele arquivo ou, pior ainda, como no caso da declaração do imposto de renda, fomentando a pirataria por iniciativa do próprio governo. Nesse caso, ao se adotar um produto aberto ou

proprietário, a interoperabilidade deve ser uma premissa", explica. "É recomendável uma plataforma aberta, o que é diferente do *software* de código aberto. A plataforma aberta é aquela que interopera com várias outras."

Em algumas áreas de TI, como a de bancos de dados, a interoperabilidade já ocorre, num processo espontâneo de ajuste do mercado. Por exemplo, é possível "conversar" Java, PHP, Microsoft, independentemente do sistema operacional ou *software* utilizado. O produto Java, da Sun, foi concebido para ser uma ferramenta capaz de interoperar com qualquer sistema operacional, por meio do JVM – Java Virtual Machine – que funciona como intérprete do código.

Segundo o especialista em tecnologia Java, Michael Yuan (veja artigo nesta edição), é possível, por exemplo, executar uma aplicação baseada em Java em um PC, Windows, Linux, um computador Mac, uma estação Unix ou até mesmo em pequenos dispositivos como telefones celulares e computadores de mão.

No *desktop*, no entanto, a questão é mais complexa. Segundo Helvécio Borges, a Microsoft

não se enquadra nesses padrões de interoperabilidade. "Comunidades do mundo livre têm desenvolvido, através de engenharia reversa e grande esforço, funcionalidades que possibilitam ler e gravar arquivos em formatos proprietários. A livre concorrência aconteceria de fato e sem aprisionamento tecnológico se os produtos proprietários usassem formatos abertos, conhecidos e documentados, como faz o OpenOffice", constata.

"Da mesma forma, homogeneizar é uma idéia estúpida; qualquer pessoa usa roupas diferentes para situações diferentes. Por que então o mesmo padrão de *software* para todos? Algumas pessoas precisam do Word, porque elaboram documentos complexos; outras usam pouco os recursos do editor, podem ter um *software* mais simples e

mais barato. Há casos também em que você não encontra a solução de que precisa no *software* livre; busca-se então o proprietário", conclui.

Essa liberdade de escolha está vinculada principalmente à necessidade de exclusividade de produção do usuário de *software*. O uso de um *software* proprietário para desenvolvimento de efeitos especiais em um filme é um exemplo. Se dois produtores adquirem o mesmo *software* comercial, com o código fechado, as possibilidades de inovação são as mesmas para os dois, não há espaço para um diferencial; no caso do código aberto, o artista pode desenvolver recursos diferentes. Trata-se da vantagem tecnológica estratégica. Os efeitos especiais do filme Titanic, por exemplo, foram desenvolvidos em *software* livre."

## Dossiê Entrevista Dossiê Entrevista

### Entrevista com o professor *Rubens Queiroz,* *da Unicamp*



Divulgação

*O criador do site Dicas-L, professor Rubens Queiroz de Almeida, fala das tendências do mercado de informática no Brasil com o crescimento do uso da internet e da conseqüente consolidação do desenvolvimento colaborativo. Ele é gerente do Centro de Computação da Unicamp, analista de sistemas, especialista em Linux, Unix e derivados. O Dicas-L, que reúne hoje mais de 27 mil assinantes, oferece informações de interesse para profissionais de informática que atuam com sistemas livres como Linux, FreeBSD e softwares associados. É autor dos livros "Linux – Dicas e Truques", "As Palavras Mais Comuns da Língua Inglesa" e "Read in English – Uma Maneira Divertida de Aprender Inglês" e colaborou na confecção do livro "A Leitura nos Oceanos da Internet".*

*Seu trabalho com o Dicas-L e o site <http://www.idph.net/> reflete a filosofia de compartilhamento de informações: quais as estatísticas de assinantes, dicas e outros números que podem ilustrar a aceitação dessa iniciativa? Qual o número de acessos mensais no Instituto de Desenvolvimento do Potencial Humano?*

A Dicas-L reuniu, em 14 de abril de 2005, 27.241 assinantes. A média de crescimento é por volta de 15 a 20 assinantes por dia. O *website* recebe em média entre 2.000 e 3.000 visitantes diariamente. O *site* do IDPH recebe, também em média, cerca de 700 visitantes/dia.

*Na sua opinião, quais as tendências no mercado de informática para produtos abertos e proprietários? A médio e longo prazos, como o mercado vai se comportar? Qual é, na sua opinião, o futuro para as informações totalmente compartilhadas sem qualquer custo financeiro?*

A internet e o baixo custo dos computadores têm sido os principais fatores na disseminação do *software* livre. A comunicação global oferecida pela internet permite que as pessoas compartilhem seus conhecimentos, projetos, idéias. A atividade de programação de computadores é extremamente gratificante e criativa e pode ser considerada um passatempo da melhor qualidade. Programadores e projetistas de *software* se dedicam ao movimento de *software* livre por diversas razões, mas o motor principal desse movimento é o reconhecimento de seu trabalho e genialidade. Dentro dessa perspectiva, onde o produto intelectual de uma pessoa ou mais pessoas pode ser amplamente distribuído através da internet, o modelo de comercialização de *software* proprietário certamente terá de ser repensado. Vejamos o caso de uma universidade, que precisa de uma quantidade enorme de sistemas para o seu funcionamento. O modelo de *software* livre é extremamente interessante pois promove a cooperação entre universidades com necessidades semelhantes, ao mesmo tempo em que permite o desenvolvimento de soluções criativas para os mais diversos tipos de questões.

*Como o brasileiro assimila as peculiaridades do software de código aberto e das informações totalmente compartilhadas? Há resistências? Em que setores isso é mais evidente? Qual a receptividade nas iniciativas pública e privada?*

Eu acho que mesmo hoje, em que o *software* livre está bem mais difundido, ainda há um grande desconhecimento dos valores filosóficos do movimento de *software* livre. A questão essencial não é preço e sim liberdade. Colocar todos os negócios de uma empresa em cima de um sistema totalmente fechado é loucura. É óbvio que, quanto maior a dependência, maior a liberdade do fornecedor de fazer o que quiser, ao preço que achar



*A atividade de programação de computadores é extremamente gratificante e criativa...*

conveniente. Mas poucos param para pensar sobre essa ótica. Basicamente, segundo a Free Software Foundation, todo *software* deve ter quatro liberdades básicas: a liberdade de se executar o programa para qualquer finalidade, a liberdade de modificar o programa para se adequar às suas necessidades, a liberdade de distribuir cópias do programa, com o código-fonte, e a liberdade para se criar uma comunidade, permitindo o aperfeiçoamento do programa ao mesmo tempo em que se ajuda o próximo. As resistências se encontram em todos os lugares e basicamente refletem o temor do novo, o apego a velhas práticas, mesmo que as novas possam trazer grandes benefícios. No momento atual, em que o

governo federal adotou a bandeira do *software* livre, nós vemos diversas iniciativas acontecendo por todo o país. Nas empresas as coisas ocorrem de forma mais silenciosa, mas não há como negar a popularização do *software* livre. O custo de se manter um parque computacional totalmente proprietário está se mostrando inviável e os riscos de se manter esse mesmo parque com *software* ilegal é um risco cada vez maior.

***Afinal, democratizar conhecimentos fere o direito autoral?***

Toda a nossa civilização foi erguida através do compartilhamento de informações. A internet



*As resistências se encontram em todos os lugares e basicamente refletem o temor do novo, o apego a velhas práticas...*

trouxe modificações radicais ao mundo como o conhecemos. O setor de música e filmes está sofrendo um baque tremendo com a emergência de tecnologias de compartilhamento descentralizadas (*peer-to-peer* ou p2p). Em busca da defesa de seus interesses, essas indústrias estão tentando reverter uma característica que permitiu a evolução da espécie humana, geração após geração, construindo sobre os conhecimentos criados anteriormente. O fato de um conteúdo estar na internet não retira os direitos do autor. A lei do direito autoral garante a propriedade de autoria de um documento desde o momento em que é publicado, independentemente de haver ou não o registro prévio.

***Quais os reflexos da internet na produção cultural e de software?***

Muito positivos. A internet oferece a todos a oportunidade de publicar seu trabalho, disseminar suas idéias, criar comunidades. É claro que isso tem seu lado bom e ruim. A primeira objeção levantada quanto a essas facilidades é justamente a qualidade do que se publica. No processo tradicional de publicação, através de editoras, geralmente o controle de qualidade é melhor, os trabalhos passam por diversas revisões e assim por diante. O lado perverso do esquema tradicional de publicação é justamente o fato de acrescentar à questão da qualidade o fator de mercado. Se não há perspectivas boas de venda, não se publica. A internet elimina a barreira de entrada e o que é bom sem dúvida irá se sobressair. Eu tenho três livros publicados, e todos esses livros foram criados e publicados primeiramente na internet para só então serem impressos e comercializados pelos canais normais.

***Como é o mercado internacional de software livre? Quais as perspectivas para o Brasil?***

Esse mercado ainda está descobrindo o seu caminho, desenvolvendo modelos de negócios viáveis. Existem empresas, como a Cygnus, que foi comprada alguns anos atrás pela RedHat, que sempre trabalharam com o modelo de *software* livre e se deram muito bem. No Brasil a iniciativa de maior destaque é a Solis (<http://www.solis.coop.br>), que nasceu dentro da Univates, em Lajeado, RS, e que se estabeleceu como uma cooperativa que hoje presta serviços para empresas e instituições em todo país. São 50 pessoas vivendo exclusivamente da prestação de serviços com *software* livre.

***Fale sobre experiências do setor público no desenvolvimento de software livre.***

Toda iniciativa que parte do Estado é complicada devido ao fator político. As iniciativas geralmente morrem ou são abandonadas com a mudança de governo. Entretanto podemos citar alguns exemplos bastante positivos, derivados da iniciativa do

governo do Rio Grande do Sul, de adotar *software* livre. O Fórum Internacional de Software Livre, que entra em 2005 na sua sexta edição, é hoje, já desvinculado de qualquer governo, um dos maiores eventos do mundo em sua especialidade. Milhares de pessoas já foram ao Rio Grande do Sul aprender o que é *software* livre, aprender sobre suas vantagens e levar de volta para sua cidade o que aprenderam, criando novos movimentos. O Estado, por seu tamanho, tem um poder enorme. Mesmo com as transições de governo, na questão de *software* livre, algo de bom sempre fica.

***O poder econômico pode ser um entrave à disseminação do conhecimento e ao desenvolvimento?***

Sim, claro. É um assunto muito amplo, mas eu gostaria de recomendar a leitura do livro "Free Culture", de Lawrence Lessig, que descreve de forma muito didática e clara como isso está ocorrendo nos dias atuais. Esse livro, disponibilizado na internet gratuitamente, já foi traduzido para o português e pode ser baixado da biblioteca digital de *software* livre da Unicamp (<http://www.rau-tu.unicamp.br/nou-rau/softwarelivre>). Resumidamente, a internet trouxe mudanças radicais na forma como conteúdo é criado e distribuído. Isso é uma ameaça para empresas que ainda não conseguiram adequar seu modelo de negócios aos novos tempos e que só conseguem enxergar um futuro em que as liberdades dos usuários sejam cerceadas para manter seus antigos privilégios. As leis criadas para combater a assim chamada "pirataria" são um exemplo claro disso. Mas como diz o próprio Lessig, qualquer lei que torne criminosa a maior parte da população de um país é uma lei errada. Quem nunca baixou uma música da internet?

***As questões jurídicas estão de acordo com as novas exigências impostas pela tecnologia?***

Isso tem sido um aspecto complicado. No meu entendimento, os avanços nessa área têm sido muito lentos. É importante que a sociedade se

mobilize intensamente para garantir que leis que vão contra seu interesse não sejam aprovadas. Mas é uma luta difícil que requer atenção constante.

***Como o mundo virtual tem interferido no mundo real?***

Os efeitos são inúmeros. O que eu considero mais importante é justamente o fato da rede dar voz a todos. Com a voz se tem poder. Quem poderia pensar que um programador solitário, chamado Linus Torvalds, iria iniciar um movimento tão forte como o do Linux, que hoje é uma ameaça real a diversas empresas monopolistas e arrogantes do topo de sua montanha de dinheiro? Como Linus Torvalds exis-



***Perverso no esquema tradicional de publicação é justamente o fato de acrescentar à questão da qualidade o fator de mercado.***

tem dezenas de outros, aparecendo diariamente com idéias e conceitos novos, que podem revolucionar a maneira como pensamos e vivemos.

***Qual é/será o papel das instituições educacionais na formação e mudança de paradigmas que envolvam a democratização do conhecimento? O modelo americano, muito adotado no Brasil, é, na sua opinião, contrário ao movimento de compartilhamento total das informações e conhecimentos? Como agilizar o processo de divulgação e compartilhamento de informações no Brasil?***

O modelo educacional certamente deve ser repensado. A internet chega a todos os lugares.

Até hoje, para estudar em uma universidade de renome, o aluno tem, em muitos casos, de se mudar para a cidade onde a universidade se encontra, arcar com despesas de aluguel, alimentação, transporte, etc., despesas que estão fora do alcance da maioria da população. Uma seleção perversa. Talentos que o país poderia utilizar para seu desenvolvimento ficam dessa forma à margem de nossa sociedade, que valoriza o diploma antes de tudo. Espero que gradualmente as universidades públicas brasileiras venham a oferecer oportunidades a mais pessoas através do ensino a distância. A Unicamp já deu um passo importante nessa direção através do Portal Ensino Aberto, que publica na internet o material didático de disciplinas de graduação. É ainda um projeto experimental, mas espera-se que ao longo

dos próximos anos uma quantidade cada vez maior esteja aberta para a população. Em uma segunda etapa, quem sabe um dia os próprios cursos de graduação sejam ministrados a distância? Um outro exemplo importante é a biblioteca digital de teses e dissertações da Unicamp (<http://libdigi.unicamp.br>), que publica hoje mais de 4.000 trabalhos acadêmicos. Já foram feitos, em cerca de três anos de funcionamento, downloads de aproximadamente meio milhão de documentos. O *software* da biblioteca, o sistema Nou-Rau, é livre e gratuito e certamente pode facilitar a adoção de estratégias semelhantes por outras universidades. É gratificante saber que muitas universidades e instituições possuem programas semelhantes, de abertura do conhecimento. Cabe à nossa sociedade encorajar e valorizar essas iniciativas.

Dossie Entrevista Dossie Entrevista Dossie Entrevista

Isabela Abreu



Márcio Girão, da Softex

## *Pesquisa Softex/Unicamp revela panorama do Software Livre no Brasil*

É de R\$ 77 milhões o mercado estimado de *software* livre hoje no Brasil, considerando-se apenas os serviços Linux. A informação foi divulgada no final do mês de abril deste ano, como resultado da maior pesquisa individualizada por país já realizada no mundo: "O Impacto do *Software* Livre e de Código Aberto na Indústria de Software do Brasil". O estudo, feito pelo Observatório Econômico da Softex (Associação para Promoção da Excelência do Software Brasileiro) e o Departamento de Política Científica e Tecnológica da Unicamp, com apoio do Ministério da Ciência e Tecnologia, estima ainda que há cerca de 195 mil servidores e 561 mil *desktops* operando com Linux no Brasil; 12% dos servidores instalados são Linux, contra 60% Windows. A previsão de crescimento para o setor é de 2,5 a 3 vezes até 2008, alcançando algo entre R\$ 192 e R\$ 231 milhões. O não pagamento de licenças poderia significar uma economia de cerca de R\$ 85 milhões/ano.

A pesquisa revela ainda um processo significativo de profissionalização dos desenvolvedores e empresas, especialmente nos últimos cinco anos, com a criação do Open Source Developer Labs (OSDL), uma organização sem fins lucrativos financiada por grandes empresas como a IBM, Intel e HP, para desenvolvimento

do Linux para ambientes de grande escala de produção. A maioria dos profissionais especializados está, hoje, contratada pelas grandes corporações e tem um perfil semelhante ao do europeu.

No caso das empresas que atuam com *software* livre, foram identificados três perfis distintos: as pequenas e médias, fundadas nas décadas de 80 e 90, que atuam principalmente com *software* proprietário, mas que começam a dedicar uma pequena parte de suas atividades a essa alternativa devido às exigências do mercado; pequenas e médias empresas fundadas mais recentemente, em grande parte devido ao código aberto, que têm a maior parte de suas atividades nesse modelo; e grandes empresas, algumas multinacionais, que também ingressaram recentemente no mundo de código aberto com estratégias específicas. As empresas estão localizadas principalmente nas regiões sul (29%) e sudeste (53%).

Uma das iniciativas da pesquisa foi o levantamento de dados através de entrevistas com empresas desenvolvedoras e fornecedoras de *software* livre. Entre as cinco representantes da área pública, nas esferas municipal e federal, constatou-se que todas possuem um projeto de migração de plataformas baixas das áreas administrativas para soluções em *software* livre. Um grande desafio dessas empresas é lidar com os sistemas legados, cuja conversão custaria grandes investimentos.

Tomando-se o uso dos sistemas operacionais em servidores por tipo de aplicação, nota-se a clara predominância do Windows, especialmente nas funções de gerenciamento de rede e banco de dados. Em segundo lugar está o Linux, com destaque nas aplicações para servidor de correio, web e gerenciamento da segurança.

Entre as empresas usuárias, 64% têm faturamento superior a R\$ 1 milhão/ano (48% acima de R\$ 50 milhões) e 65% possuem mais de 99 funcionários. O estudo revelou ainda as quatro maiores motivações para a adoção do *software* de código aberto. Os desenvolvedores apontaram questões como desenvolver novas habilidades, compartilhar conhecimento, resolver problema sem solução com o *software* proprietário, melhor empregabilidade e aplicações comerciais. Já para os usuários, a motivação passa pela redução de custos, desenvolvimento de novas habilidades, facilidade das ferramentas e justificativas ideológicas, que, segundo a pesquisa, são mais fortes entre os usuários do que entre os desenvolvedores.

O levantamento dos dados incluiu um painel de especialistas, uma enquete eletrônica com 3.657 respondentes, um conjunto de entrevistas com empresas desenvolvedoras e usuárias de *software* de código aberto e um levantamento de informações secundárias sobre empresas que lidam com *software* livre e *open source* no Brasil. O trabalho está disponível em [www.softex.br](http://www.softex.br).

## *A experiência do Governo Federal*

---

Com a expectativa de economizar nos próximos três anos 80% do valor pago atualmente com o uso de licenças de *software*, o Governo Federal empreende, desde outubro de 2003, um projeto amplo de migração de todas as suas aplicações para *software* de código aberto. Os argumentos misturam questões técnicas, econômicas e ideológicas:

maior segurança das informações, estabilidade dos sistemas, independência tecnológica, disseminação do conhecimento técnico, redução de custos e viabilização do programa de inclusão digital.

Atualmente não há legislação que obrigue a adoção do *software* livre pelos órgãos do governo federal: há apenas uma orientação do ITI - Instituto

Nacional de Tecnologia da Informação, autarquia vinculada à Casa Civil. Mas um Decreto que regulamentará a obrigatoriedade do uso já está sendo analisado. Após a edição do decreto, para aquisição de *software* proprietário os ministros deverão apresentar argumentação justificando a escolha.

Criado em outubro de 2003, o Comitê Técnico de Implantação do Software Livre atua no âmbito do Comitê Executivo do Governo Eletrônico, coordenado pelo ITI. O processo de migração se fundamenta no Guia Livre – Referência de Migração para Software Livre, editado em português e espanhol e disponível no endereço [www.governoeletronico.gov.br](http://www.governoeletronico.gov.br).

Segundo o gerente de Inovações Tecnológicas da Secretaria de Logística e Tecnologia da Informação (SLTI) do Ministério do Planejamento, Corinto Meffe, as ações previstas no processo de migração têm resultado no combate a mitos que prejudicam a aceitação do *software* livre: o mito do treinamento, por exemplo – de que não há treinamento e capacitação – foi contraposto com a ação do ITI, que com diversos outros órgãos, organizou a semana de capacitação, que resultou em duas mil pessoas treinadas concomitantemente, em 150 cursos e 180 profissionais envolvidos. "Demonstramos que existem profissionais capacitados, que existe treinamento."

Isabela Abreu



Corinto Meffe, do Ministério do Planejamento

*O gerente de Inovações Tecnológicas da SLTI, Corinto Meffe, falou à Revista Fonte sobre a experiência de migração para o software livre no Governo Federal.*

**Qual a avaliação do impacto do Guia Livre – Referência de Migração para o Governo Federal?**

O Guia Livre tornou-se referência para a organização das migrações em instituições públicas e privadas. Alcançamos a marca de mais de 40.000 *downloads*

A adoção do *software* livre pelo governo federal como política pública naturalmente suscitou reações. O presidente da Sun Microsystems, Jonathan Schwartz, publicou carta aberta ao presidente do Brasil, em seu *blog*, aplaudindo a adoção do modelo, "Que permite que todos os setores da população, e não apenas os privilegiados e influentes, tenham acesso à auto-determinação e independência. Uma rede aberta e *software* livre são a base para oportunidades iguais e o seu compromisso é um alerta para o mundo ver que o Brasil pretende desenvolver suas próprias soluções tecnológicas, suas próprias competências e sua própria indústria".

A Sun fatura anualmente US\$ 12 bilhões, está presente em mais de 60 países. Responde por 35% em plataformas de grande porte no Brasil e trabalha com a maior comunidade de desenvolvedores.

Já o presidente da Microsoft Brasil, Emilio Umeoka, afirmou em entrevista à publicação Executivo de Valor (abril/05) que "Temos buscado manter um grau de diálogo muito aberto com o governo, nos concentrando na área técnica, sem discutir questões ideológicas. O debate que o governo ou qualquer empresa privada tem que focalizar é a relação custo/benefício e o retorno sobre investimento. No caso específico do governo, é como tornar a máquina pública mais eficiente."

e um retorno positivo dos usuários que mantiveram contato direto com o documento. Percebemos que ajudamos principalmente as prefeituras, governos estaduais e pequenas empresas, que atualmente usam o documento como norteador para a estruturação de seus planos e suas ações de migração. Importante ressaltar que o documento tem versão em espanhol, o que nos possibilitou a entrega oficial para os governos da Argentina, Uruguai, Chile e Venezuela, além da repercussão na Comunidade Européia (fonte primária do documento brasileiro). Com base no Guia Livre também foi elaborado o plano de migração para *software* livre do Ministério do Planejamento, em curso desde o início do ano.

***Quais as estratégias de migração existentes e qual foi a adotada pelo Ministério do Planejamento?***

Existem as mais variadas formas para se começar uma migração. Isso ocorre principalmente em função dos ambientes heterogêneos de cada organização. As migrações podem começar pelos servidores, pelas estações de trabalho, pelo uso de emuladores, pelas aplicações ou em novas tecnologias e sistemas. Mesmo algumas dessas alternativas de rotas de migração podem conter grupos menores de atuação, como no caso dos servidores, que podem ser de arquivos, de comunicação, de aplicação e de internet, ou no caso das estações, para ferramentas de automação, os navegadores, o correio "cliente" e até o sistema operacional. O Ministério do Planejamento estruturou uma ação fortemente concentrada no diagnóstico, ou seja, uma fase de levantamento prévio do ambiente. Dessa forma, pretende-se aplicar qualquer um dos caminhos/rotas descritos acima. A proposta é aplicar a melhor solução para o ambiente e o usuário, de acordo com o que foi percebido na fase de diagnóstico.

***Qual é o principal ponto de sustentação desse plano?***

Seria o mapeamento da estrutura de informática e serviços existentes para dimensionar

corretamente a melhor forma de migração, sem dificultar o dia-a-dia do usuário. Estabelecemos esse primeiro ciclo de diagnóstico onde se realiza um mapeamento de todos os sistemas utilizados. Depois desse levantamento, passamos as informações para um banco de dados que nos sinaliza quais os melhores usuários para migrarmos. Essa escolha acontece com os seguintes critérios: aplicativos com similar livre, menor quantidade de aplicativos ativos, menor microlegado, menor macrolegado, etc. O nosso aprendizado da migração já começou com os grupos de voluntários: pessoas que se apresentaram como interessados pelo uso de *software* livre no seu ambiente de trabalho, que servem de base para aferirmos os impactos e as dificuldades. Fazemos um relatório e avaliamos frente às outras experiências e tendo como base o diagnóstico descrito anteriormente. Buscamos com a conjugação dessas duas macroações – migração dos voluntários e diagnóstico – deixar o ambiente propício para a migração em larga escala. Considero que o ponto forte do plano de migração seja justamente o mapeamento do ambiente e a minimização do impacto para o usuário.

***Como essa estratégia pode ajudar os demais órgãos?***

Por ser tratar de um plano com base nas técnicas do PMI (Project Management Institute), que usa ferramentas livres de gerenciamento de projetos e é fortemente documentado, o nosso documento pode servir de referência para elaboração de planos adequados a cada instituição, o que já vem acontecendo na ANTT – Agência Nacional de Transportes Terrestres – e no Ministério dos Esportes. É importante reforçar que não bastam bons planos; apoios gerenciais consistentes e coordenadores e técnicos capazes são fundamentais para não deixar um belo plano no papel. Muitas vezes um documento bem-estruturado serve como alento e contemplação. Migração é necessariamente "mão na massa".

## Democratização do conhecimento e o trabalho do Creative Commons

---

"Conhecimento só é gerado em liberdade. A história da universidade na civilização ocidental é a história da luta pelo estabelecimento da primazia da razão sobre a força do dogma." A frase é do professor Ulisses Leitão, diretor de Tecnologia do Instituto Doctum. Para ele, o método científico foi instaurado ao se "abrir o código das afirmações dogmáticas".

A discussão das liberdades relativas ao uso e distribuição da produção intelectual transcende o campo da tecnologia da informação e ganha proporções mundiais, contemplando todas as formas de criação intelectual. Para o diretor-presidente da Softex, Márcio Girão Barroso, "A questão do *software* livre está nos levando a pensar a propriedade intelectual de forma mais ampla. Por que, por exemplo, três ou quatro gravadoras dominam a produção fonográfica, determinando o que devemos ouvir? Essa situação começa a incomodar a sociedade; está havendo uma reação filosófica, psicológica, na verdade sociológica, a esse domínio".

Polêmicas em torno da questão de se restringir ou ampliar os direitos relacionados à produção cultural e científica resultaram na criação de um movimento mundial, nascido nos Estados Unidos e que encontrou repercussão no Brasil, que figura em destaque no trabalho do Creative Commons. Trata-se de uma organização sem fins lucrativos, concebida em 2001 na Universidade de Stanford (EUA), pelo professor Lawrence Lessig, e que tem como diretor no Brasil o advogado e professor da Fundação Getúlio Vargas, Ronaldo Lemos (veja entrevista na seção Diálogo, nesta edição).

O trabalho do Creative Commons propõe justamente uma flexibilização dos direitos autorais de "todos os direitos reservados" para "alguns direitos reservados", criando licenças que dão suporte legal a autores que desejam permitir aos consumidores de suas produções mais do que o determinado pelo tradicional *copyright*. Um dos grandes

adeptos do movimento, o ministro da Cultura, Gilberto Gil, publicou sob licença Creative Commons (CC) uma de suas músicas durante o Fórum Mundial realizado em Porto Alegre, em 2004.

O professor Ulisses Leitão, do Instituto Doctum, defende a idéia de que a revolução científica tem a sua origem na liberdade de conhecimento. "Nesse sentido, não é sem razão que o iluminismo ocorre concomitante com a invenção da imprensa por Hans Guttemberg, em 1436. Pois além de se estabelecer o direito à informação, há que se estabelecer os meios tecnológicos para a sua propagação. A revolução da tecnologia da informação só está em andamento porque os grandes cientistas que estabeleceram a base científica para essa revolução sempre liberaram de forma livre as suas idéias em benefício de toda a humanidade. Dentro desse contexto, creio que *software* livre é a única base possível para a sociedade do conhecimento, que é a agenda da Humanidade nos dias atuais."

Ao comentar a liberdade de propriedade da produção intelectual, o professor Ulisses fala sobre a questão das patentes de *software*. Para ele, a atual discussão de propriedade intelectual sofre influências do modelo americano. "Acelerada a partir de 1999, há uma corrida pelo registro de patentes de *software* nos Estados Unidos. São realizados registros de patentes óbvias, como "Save As" e "Sistema de busca no site". Segundo o professor, a agência de patentes recebe pelo registro, portanto, quanto mais pedidos de patente, melhor. "Não há condições operacionais de avaliação criteriosa da solicitação de patente e muito menos há interesse em se limitar as patentes, pois elas formam a base da hegemonia econômica americana", conclui. Ulisses Leitão adverte que se o Brasil se curvar à tentativa de se colocar as patentes de *softwares* no mesmo patamar dos direitos de propriedade intelectual, o país perderá a oportunidade de desenvolvimento

econômico pelos próximos cinquenta anos. "A situação é tão grave que, em caso de aprovação da patente de *software*, o *software* livre será afetado, pois mesmo que se queira liberar um *software* inteiramente desenvolvido pelo autor, ele será

impedido, pois fatalmente usará alguma 'idéia óbvia' já patenteada. A luta contra a patente de *software* está em pleno andamento na Europa e temo que a sociedade brasileira não esteja devidamente preparada para esse debate".

## *A Wikipedia e a geração compartilhada de conteúdos*

A criação colaborativa tem filhotes fora do mundo dos especialistas desenvolvedores de *software*. Um exemplo de sucesso de produção coletiva do conhecimento é a Enciclopédia Wikipedia - [www.wikipedia.com](http://www.wikipedia.com) – escrita em cerca de 80 idiomas e alimentada pela colaboração de voluntários.

Criada em setembro de 2001 e gerida pela Fundação Wikimedia – organização sem fins lucrativos –, a enciclopédia está disponível na internet para quem quiser consultá-la gratuitamente ou contribuir voluntariamente com informações. Qualquer conteúdo pode ser copiado e modificado livremente.

O projeto é baseado na filosofia wiki, ferramenta para a geração coletiva de conteúdos, que podem ser publicados automaticamente, sem uma revisão prévia. Dessa forma, vários colaboradores podem trabalhar simultaneamente num mesmo conteúdo, que passa a integrar a enciclopédia. Qualquer pessoa pode se cadastrar no sistema e publicar artigos ou escolher temas que aguardam um editor. O pressuposto é de que a comunidade é capaz de fiscalizar o conteúdo disponível para os leitores, eliminando erros e melhorando a qualidade das informações.

### Glossário

#### *O que é software livre?*

(extraído do site do Governo Federal - [www.softwarelivre.gov.br/SwLivre/](http://www.softwarelivre.gov.br/SwLivre/))

O texto a seguir, com a definição de *software* livre, é de autoria do Prof. Roberto Hexsel do Departamento de Informática da Universidade Federal do Paraná. Para baixar o texto completo, intitulado "Propostas de Ações de Governo para Incentivar o Uso de Software Livre", acesse o endereço <http://www.inf.ufpr.br/~roberto/public.html>.

**BSD** - A licença BSD cobre as distribuições de *software* da Berkeley Software Distribution, além de outros programas. Essa é uma licença considerada 'permissiva' porque impõe poucas restrições sobre a forma de uso, alterações e redistribuição do *software* licenciado. O *software* pode ser vendido e não há obrigações quanto à inclusão do código-fonte, podendo o mesmo ser incluído em *software* proprietário. Essa licença garante o crédito aos autores do *software* mas não tenta garantir que trabalhos derivados permaneçam como *software* livre.

**Copyleft** - A maioria das licenças usadas na publicação de *software* livre permite que os programas sejam modificados e redistribuídos. Estas práticas são geralmente proibidas pela legislação internacional de *copyright*, que

tenta justamente impedir que alterações e cópias sejam efetuadas sem a autorização do autor. As licenças que acompanham *software* livre fazem uso da legislação de *copyright* para impedir utilização não-autorizada, mas essas licenças definem clara e explicitamente as condições sob as quais cópias, modificações e redistribuições podem ser efetuadas, para garantir as liberdades de modificar e redistribuir o *software* assim licenciado. A esta versão de *copyright* dá-se o nome de *copyleft*.

**Debian** - A licença Debian é parte do contrato social celebrado entre a Debian e a comunidade de usuários de *software* livre e é chamada de Debian Free Software Guidelines (DFSG). Em essência, essa licença contém critérios para a distribuição além da exigência da publicação do código-fonte. Esses critérios são: (a) a redistribuição deve ser livre; (b) o código-fonte deve ser incluído e deve poder ser redistribuído; (c) trabalhos derivados devem poder ser redistribuídos sob a mesma licença do original; (d) pode haver restrições quanto à redistribuição do código-fonte se o original foi modificado; (e) a licença não pode discriminar qualquer pessoa ou grupo de pessoas, nem quanto a formas de utilização do *software*; (f) os direitos outorgados não podem depender da distribuição onde o *software* se encontra; e (g) a licença não pode "contaminar" outro *software*.

**Freeware** - O termo *freeware* não possui uma definição amplamente aceita, mas é usado com programas que permitem a redistribuição mas não a modificação e seu código fonte não é disponibilizado. Esses programas não são *softwares* livres.

**GPL** - A Licença Pública Geral GNU (GNU/General Public License) é a licença que acompanha os pacotes distribuídos pelo Projeto GNU e mais uma grande variedade de *softwares*, incluindo o núcleo do sistema operacional Linux. A formulação da GPL é tal que, ao invés de limitar a distribuição do *software* por ela protegido, ela de fato impede que esse *software* seja integrado em *software* proprietário. A GPL é baseada na legislação internacional de *copyright*, o que deve garantir cobertura legal para o *software* assim licenciado.

**Open Source** - A licença do Open Source Initiative é derivada da licença Debian, com as menções à Debian removidas.

**Shareware** - *Shareware* é o *software* disponibilizado com a permissão para que seja redistribuído, mas a sua utilização implica o pagamento pela sua licença. Geralmente, o código-fonte não é disponibilizado e, portanto, modificações são impossíveis.

**Software Comercial** - *Software* comercial é o *software* desenvolvido por uma empresa com o objetivo de lucrar com sua utilização. Note que 'comercial' e 'proprietário' não são a mesma coisa. A maioria dos *softwares* comerciais é proprietário mas existe *software* livre que é comercial e existe *software* não-livre não-comercial.

**Software em Domínio Público** - *Software* em domínio público é *software* sem *copyright*. Alguns tipos de cópia ou versões modificadas podem não ser livres porque o autor permite que restrições adicionais sejam impostas na redistribuição do original ou de trabalhos derivados.

**Software Livre (Free Software)** - É o *software* disponível com a permissão para qualquer um usá-lo, copiá-lo e distribuí-lo, seja na sua forma original ou com modificações, seja gratuitamente ou com custo. Em especial, a possibilidade de modificações implica que o código fonte esteja disponível. Se um programa é livre, potencialmente ele pode ser incluído em um sistema operacional também livre. É importante não confundir *software* livre com *software* grátis, porque a liberdade associada ao *software* livre de copiar, modificar e redistribuir independe de gratuidade. Existem programas que podem ser obtidos gratuitamente, mas que não podem ser modificados nem redistribuídos. Por outro lado, existe a possibilidade de uso não-gratuito em todas as categorias. Há uma cópia da definição de *software* livre pela Free Software Foundation publicada na página <http://www.fsf.org/philosophy/free-sw.pt.html>

**Software Proprietário** - *Software* proprietário é aquele cuja cópia, redistribuição ou modificação são em alguma medida proibidos pelo seu proprietário. Para usar, copiar ou redistribuir deve-se solicitar permissão ao proprietário ou pagar para poder fazê-lo.

**Software Semilivre** - *Software* semilivre é *software* que não é livre, mas é concedida a permissão para que indivíduos o usem, copiem, distribuam e modifiquem, incluindo a distribuição de versões modificadas, desde que o façam sem o propósito de auferir lucros. Exemplos de *software* semilivre são as primeiras versões do Internet Explorer da Microsoft, algumas versões do *browser* da Netscape e o StarOffice.

**X.org** - O Consórcio X distribuiu o X Window System sob uma licença que o faz *software* livre mas não adere ao *copyleft*. Existem distribuições sob a licença da X.org que são *softwares* livres, e outras distribuições que não o são. Há algumas versões não-livres do sistema de janelas X11 para estações de trabalho e certos dispositivos do IBM-PC que são as únicas funcionais disponíveis, sem similares distribuídos como *software* livre.

## Bookmarks

Associação para a Promoção da Excelência do Software Brasileiro - [www.softex.br](http://www.softex.br)  
 Berkman Center for Internet & Society at Harvard Law School - <http://cyber.law.harvard.edu>  
 Celepar - [www.celepar.pr.gov.br](http://www.celepar.pr.gov.br)  
 Centro de Tecnologia e Sociedade / Direito FGV - <http://www.diretorio.fgv.br/cts/>  
 Centro de Computação da Unicamp - <http://www.ccuec.unicamp.br/ccuec>  
 Cimcorp - [www.cimcorp.com.br](http://www.cimcorp.com.br)  
 Código Livre - [www.codigolivre.org.br](http://www.codigolivre.org.br)  
 Comitê Técnico de Implementação de Software Livre do Governo Federal do Brasil - [www.softwarelivre.gov.br](http://www.softwarelivre.gov.br)  
 Comitê Técnico de Implementação do Software Livre - <http://interagir.softwarelivre.gov.br>  
 Creative Commons - [www.creativecommons.org](http://www.creativecommons.org)  
 Desenvolvimento Colaborativo em Software Livre - <http://colaborar.softwarelivre.gov.br>  
 EAC Software - <http://www.eacnet.com.br/>  
 Free Software Foundation - Fundação do Software Livre - <http://www.fsf.org/home.pt.html>  
 Fundação Jboss - [www.jboss.org](http://www.jboss.org)  
 Grupos de Software Livre GNU - <http://www.gnu.org.br/>  
 Grupo TBA - [www.tba.com.br](http://www.tba.com.br)  
 Grupo de Usuários GNU - <http://www.gnurias.org.br/>  
 Instituto Doctum - <http://www.doctum.com.br>  
 JCP - Java Community Process ([www.jcp.org](http://www.jcp.org))  
 License Company - <http://www.license.com.br/>  
 Linux - <http://www.linux.org/>  
 Microsoft do Brasil - [www.microsoft.com.br](http://www.microsoft.com.br)  
 Ministério da Ciência e Tecnologia - <http://www.mct.gov.br/>  
 MySQL - <http://www.mysql.com/>  
 Open Source Developer Labs (OSDL) - <http://www.osdl.org/>  
 Open Source Initiative - [www.osi.org](http://www.osi.org)  
 Projeto Software Livre Brasil - <http://www.softwarelivre.org>  
 Solis - [www.solis.coop.br](http://www.solis.coop.br)  
 Sun Microsystems - [www.sun.com.br](http://www.sun.com.br)

# O Bom é ser simples



Roberto Prado\*

## Quanto maior o grau de complexidade dos *softwares*, mais altos são os custos com suporte técnico e manutenção

Os investimentos das empresas em tecnologia da informação (TI) devem continuar crescendo em 2005. Espera-se que a migração dos negócios em busca de um ambiente mais dinâmico e eficiente continue regendo os aportes em TI este ano. Mas isso está longe de significar o fim da pressão que empresas em todo o mundo vêm sofrendo para reduzir seus orçamentos com tecnologia. Pelo contrário. É consenso no mercado que os CIOs enfrentarão cada vez mais o desafio de planejar e aplicar corretamente os recursos que dispõem. É por isso que a facilidade de implementação, a compatibilidade, a confiabilidade e o suporte técnico da tecnologia escolhida são questões que fazem diferença, afetando diretamente o custo – e o sucesso – do projeto.

Baseadas na expectativa de que o sistema operacional Linux seja gratuito, algumas empresas têm adotado distribuições comerciais desse *software* em seus servidores, com a expectativa de terem menores custos. Normalmente, esse tipo de decisão leva em conta apenas o preço de aquisição do *software* e acaba deixando de lado outros componentes que entram na conta total de um orçamento de TI e que, na verdade, representam a maior parte dos gastos. É o que comprova um recente estudo realizado pela BearingPoint. Ao simular a infra-estrutura de uma empresa de porte médio – com 29 servidores e 232

usuários –, a BearingPoint traçou um comparativo entre um cenário utilizando o Microsoft Windows Server 2003 e dois outros usando distribuições Linux – no caso Red Hat Enterprise Linux e SUSE Linux. Os resultados mostram que, devido à sua simplicidade de manutenção, o Windows é a opção mais barata quando tomado em comparação um período de utilização de cinco anos.

A vantagem da plataforma Windows também pode ser atribuída aos contínuos esforços da Microsoft em reduzir a complexidade de suas tecnologias e, assim, facilitar a vida dos administradores de TI. A abordagem centrada na simplicidade dos sistemas também garante economia para as empresas na hora de realizar tarefas de desenvolvimento e implantação de sistemas. Em testes realizados de forma independente, a plataforma Microsoft mostrou que é significativamente melhor que estruturas formadas por Linux em ambiente Java (com plataforma J2EE e IBM WebSphere 5.1). O estudo, realizado pela The Middleware Company (TMC) colocou duas equipes de desenvolvedores, uma para trabalhar na plataforma J2EE e outra com Microsoft .NET 1.1 rodando em Microsoft Windows Server 2003. Os dois times receberam a mesma especificação para construir, implementar e testar uma aplicação web em um laboratório monitorado e auditado. A equipe .NET foi mais ágil na execução de todas as tarefas de

desenvolvimento. A instalação e configuração de aplicativos no servidor, por exemplo, consumiu quatro horas/homem contra 22 horas/homem do time WebSphere.

O próprio mecanismo descentralizado do Linux favorece a dispersão de esforços, a redundância e a proliferação de iniciativas diferentes. Há centenas de versões e distribuições lançadas ao longo dos últimos anos (como mostra o site [www.levenez.com/unix](http://www.levenez.com/unix)). Em pesquisa realizada pela InformationWeek nos Estados Unidos, o problema de incompatibilidade entre os *softwares* mostrou-se a principal preocupação dos usuários de Linux (apontada por mais de 30% daqueles que adotaram o sistema em servidores e mais de 40% dos usuários em *desktops*). A falta de conhecimento técnico, a falta de documentação adequada e a diversidade de distribuições e versões aparecem também como fatores de grande preocupação.

A proliferação de versões costuma gerar problemas diversos. Se uma empresa adota uma distribuição comercial do Linux, provavelmente ficará diante de um dilema. Se atualizar o sistema de forma independente, baixando os *upgrades* diretamente da Internet, perderá o direito a suporte técnico do distribuidor. Mas, se aguardar os *updates* periódicos do fornecedor, corre o risco de conviver com incompatibilidades entre o sistema operacional e os programas desenvolvidos pelos diversos projetos de *software* de código aberto espalhados pelo mundo, que também são atualizados com grande frequência.

Para as equipes de TI das empresas, que ao final precisam de soluções ágeis e rápidas e que mantenham o foco no negócio principal da companhia, todas essas questões devem ser muito bem pesadas na hora de uma decisão. ■

\* Roberto Prado é gerente de Estratégias de Mercado da Microsoft Brasil



**A Chave da Comunic@ção Segura**

Relacionamento ágil e seguro com governos, clientes e fornecedores pela Internet. Menos burocracia, mais eficiência nos processos e redução de custos operacionais.

**OPÇÕES**  
Pessoas Físicas e Jurídicas  
- Assinatura Digital  
- Sigilo (criptografia)  
Pessoas Jurídicas  
- Servidores Web (site seguro)

**CONTATOS**  
(31) 3339-1505  
[cdigital@prodemge.gov.br](mailto:cdigital@prodemge.gov.br)  
rua Gonçalves Dias, 201  
Funcionários - CEP 30140-090  
Belo Horizonte (MG)

ICP Brasil  
ERTISIGN

AUTORIDADE CERTIFICADORA  
prodemge  
Tecnologia de Minas Gerais

GOVERNO DE MINAS  
PLANEJAMENTO E GESTÃO



# O futuro da computação

Richard Stallman\*

**É** uma tarefa impossível. Este foi o primeiro pensamento que me ocorreu quando me pediram para escrever um artigo sobre o futuro do uso dos computadores. Eu não consigo ver o futuro: a minha bola de cristal está embaçada, hoje e qualquer outro dia. Não sei o que vai acontecer.

Eu consigo imaginar vários possíveis cenários futuros para a computação na sociedade. Na realidade, o futuro será, provavelmente, diferente de todos eles. No entanto, esses cenários podem ser úteis para mostrar quais são as possibilidades abertas e as questões que devemos enfrentar. A seguir, descrevo dois cenários possíveis.

Os computadores fazem aquilo que são programados para fazer. Mas quem decide para que eles são programados? Como os computadores desempenham um papel importante na vida da maioria das pessoas, eles se tornam, cada vez mais, ferramentas com as quais algumas pessoas exercem poder sobre as outras. Existe um projeto de computação do Instituto de Tecnologia de Massachusetts (MIT) chamado "oxigênio". A idéia do projeto é que os computadores estarão por toda parte e se tornarão invisíveis. Como o oxigênio,

você não os verá mais. Eu, provavelmente, não me importaria de não ver os meus computadores desde que não os perca. Mas se não vemos o meu computador e nem o seu, também não veremos os computadores do FBI e da CIA. Essa situação é, obviamente, bastante aterrorizadora, já que o FBI é comandado pelo homem que manipulou a lei para justificar torturas, a CIA é controlada pelo homem que fingiu que os Contras não estavam criando o caos na Nicarágua e ambos são subordinados a um homem que rouba eleições, começa guerras e alega que está fazendo isso porque Deus o ordenou.

Pode não ser tão óbvio que o seu próprio computador seja tão assustador quanto os outros. Você pode até pensar que ele está sob seu controle. Mas não é o caso, se você está usando *software* proprietário. Geralmente, é proibido que os usuários compartilhem ou mudem o *software* proprietário porque eles não têm acesso ao código-fonte do programa – os planos do programa, o formato que os programadores podem entender e mudar. Somente o desenvolvedor tem o código-fonte e, por isso, ele tem o controle total sobre o que o programa faz. Os usuários podem somente aceitar ou rejeitar o *software*. Em termos

práticos, isso proporciona ao usuário um poder limitado. Certamente, você já ouviu pessoas dizerem "não tenho escolha. Tenho que usar este programa". Elas exageram. Se realmente quisessem, poderiam dizer não. No entanto, o desenvolvedor pode dominá-las quando elas acreditam que não têm escolha. Se analisarmos os fatos em conjunto, temos uma situação na qual os desenvolvedores de *software* controlam o seu computador. O *software* só lhe obedece até onde o desenvolvedor permite que isto aconteça.

Quando uma empresa tem o controle sobre o que seu *software* faz, é tentador para ela abusar desse poder. Muitos programas proprietários são desenvolvidos para coletar informações sobre você, irritá-lo ou impedi-lo de acessar informações livremente. Eles podem até ter "*back doors*"\*\* inescrupulosas. Outra possibilidade é a censura imposta pelo *software*. Estão sendo desenvolvidos programas que permitem a uma autoridade central publicar uma mensagem fazendo com que todos os computadores, inclusive o seu, se recusem a mostrar um arquivo, talvez um texto que você mesmo escreveu. Com certeza, se Bush pedisse, uma companhia norte-americana publicaria tal mensagem.

Os usuários não conseguem se livrar desses recursos maliciosos. Como não têm o código-fonte, eles não conseguem até mesmo saber quais são os recursos maliciosos que o programa contém. Tudo o que podem fazer é confiar cegamente no desenvolvedor, sem saber se ele merece ou não essa confiança.

Você pode acreditar que a concorrência limitaria o abuso de poder dos desenvolvedores. Mas sabe que uma única companhia tem quase o monopólio de muitos tipos de *software*. Mesmo quando há concorrência, isso não ajuda. Muitas vezes, todos os programas concorrentes têm recursos maliciosos semelhantes, pois os desenvolvedores têm interesses em comum. Às vezes, todos desenvolvedores assinaram acordos

para restringir os usuários da mesma maneira. Foi o que aconteceu com os DVDs, por exemplo. Você pode pensar que isso seria uma violação das leis de concorrência empresarial, mas a maioria dos governos atualmente parece deixar que isso aconteça. Eles não defendem os cidadãos do poder empresarial como deveriam.

Quando um *software* coleta informações sobre você e as envia para uma companhia nos Estados Unidos, o FBI pode acessá-las a qualquer momento, mesmo sem a autorização da Justiça. Isso faz parte do que Bush colocou na lei incorretamente chamada de Ato Patriótico dos Estados Unidos\*\*\*. Qualquer informação que o FBI não consegue obter legalmente, será provavelmente fornecida pela companhia de forma voluntária.

Felizmente, é possível ter um outro mundo do *software* e, de fato, ele já existe. É o mundo do *software* livre. Com ele, os usuários têm o controle – individualmente e coletivamente. Você pode compartilhar o *software* livre e ninguém vai acusá-lo de pirataria. Pode estudar o código-fonte e mudá-lo para fazer o que quiser. Pode também publicar a sua versão modificada do *software* para que outros possam usá-la.

Desde sua criação em 1984, a comunidade do *software* livre desenvolveu sistemas operacionais completos, como o GNU+Linux, e milhares de programas aplicativos. Na nossa comunidade de cooperação voluntária, todos os programadores podem contribuir no desenvolvimento do *software*. Simples usuários também podem contribuir. Se você não é um programador, mas quer que o programa funcione

Quando uma empresa tem o controle sobre o que seu *software* faz, é tentador para ela abusar desse poder.

de maneira diferente, pode pedir a um amigo, convencer o primo programador ou pagar a um programador de sua preferência para mudá-lo.

Quando os programadores publicam versões modificadas, você não precisa saber sobre programação para escolher qual delas gosta mais.

Quando os programadores publicam versões modificadas, você não precisa saber sobre programação para escolher qual delas gosta mais.

Ou você pode simplesmente confiar que a comunidade fará uma boa escolha. Como milhões de pessoas lêem o código-fonte do *software* livre, os recursos maliciosos podem ser facilmente identificados e removidos. A comunidade, com certeza, escolherá uma versão que não

espiona, não tem *back door*, não impõe anúncios, não se recusa a obedecer certos usuários e não deixa ninguém censurar o que você escreve.

Os desenvolvedores de *software* proprietário ficam tentados a acrescentar recursos maliciosos porque seu poder lhes permite permanecer impunes. Tal comportamento errôneo é improvável quando os usuários têm o controle.

De que maneira o futuro se desenrolará? Eu não sei, porque isto depende de você. Se sempre escolher o caminho mais fácil, os desenvolvedores de *software* proprietário (assim como muitos outros) vão tirar vantagem da sua ênfase em resultados a curto prazo para levá-lo para a direção que eles querem. Mas se exigir liberdade no seu computador, isso está agora ao seu alcance. Suas decisões e escolhas de como viver e de que tipo de vida você apóia vão determinar nosso futuro. ■

\* Em 1984, Richard Stallman lançou o desenvolvimento do sistema operacional GNU e, em 1985, fundou a Free Software Foundation (Fundação Software Livre). Desde então, ele recebeu vários prêmios e doutorados honorários.

Copyright 2005 Richard Stallman

É permitida a cópia e distribuição deste artigo na íntegra, sem modificações, em todo o mundo, através de qualquer meio, sem pagamento de *royalty*, desde que esta nota não seja retirada.

#### Notas da tradutora:

\*\* *back door*: elemento do *software* escondido em um sistema que pode ser usado para invadir a privacidade do usuário.

\*\*\* Ato Patriótico (USA Patriot Act): Lei sobre segurança nacional, promulgada depois do atentado de 11 de setembro de 2001.

Tradução: Ana Camargos, jornalista e mestre em Comunicação pela Universidade da Flórida.

*Experiências concretas de uso do software de código aberto revelam os principais desafios e benefícios de sua adoção.*

*Veja nesta seção como o Metrô de São Paulo e a Prefeitura de Recife concretizaram seus projetos. Conheça também o processo em andamento no Governo do Paraná.*



## Metrô-SP

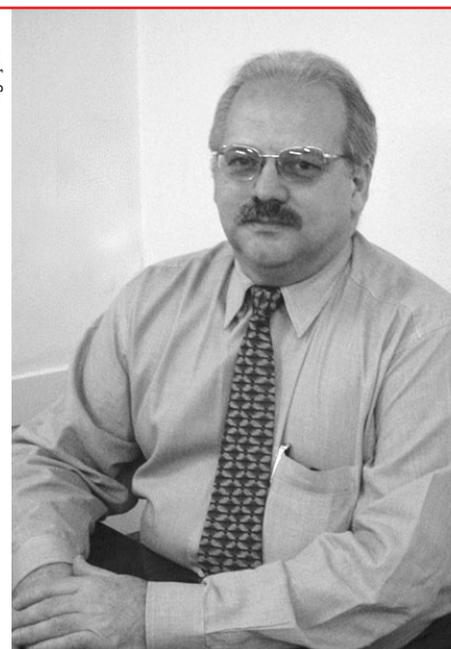
A experiência da Companhia do Metropolitano de São Paulo – o Metrô-SP – teve início em 1997, motivada pela necessidade de redução de custos, que se tornou premente em função do corte de subsídios do Governo do Estado em 1995. Segundo o gerente de Tecnologia da Informação da empresa, Gustavo Celso de Queiroz Mazzariol, o processo começou com a substituição do correio, baseado em *mainframe*, pelo MetroMail e utilizado até hoje, totalmente baseado em *software* livre. "Os custos despencaram para menos de R\$ 10 mil por mês. E o impacto dessa implantação junto aos nossos usuários foi extremamente leve, principalmente pela facilidade de uso e interface amigável", relembra.

No caso da suíte de escritório, a experiência foi diferente: "O melhor acordo obtido com a Microsoft era um contrato de US\$ 300 por micro por ano, com direito à última versão do MS-Office e Windows. Como estávamos com 1.300 equipamentos, os custos atingiriam a bagatela de US\$ 390 mil por ano ou R\$ 800 mil em reais da época. A opção identificada foi o StarOffice – de uso gratuito e rodando sob Linux - e com uma grande empresa por trás, a Sun Microsystems, que acabava de adquirir a Star Division, proprietária do produto."

Mazzariol lembra a ousadia da experiência na época, considerando não haver então *cases* conhecidos no país. "Nós mesmos não tínhamos tempo para grandes estudos e planejamentos." Foi criado então um plano para implantação progressiva acompanhada, com um trabalho de sensibilização dos funcionários por meio de palestras de divulgação e esclarecimentos itinerantes por toda a empresa, mostrando semelhanças entre o StarOffice e o MS-Office. Foram distribuídos 500 CDs do StarOffice e montado um *help-desk*, para atendimento às primeiras dúvidas.

"Em abril de 2001, por conta de uma reportagem na revista InfoExame, hoje INFO, intitulada "Estação StarOffice", acabamos virando um *case* de mercado, lembra Mazzariol. "Ficamos surpresos com a repercussão do caso. Desde então, temos

Divulgação



"Em termos de economia, atualmente temos algo em torno de R\$ 3 milhões por ano"

# Software Livre e o mercado



Isabela Abreu

Cezar Taurion\*

A IBM tem forte atuação no movimento de *software* livre. Diversas ações de peso têm sido efetuadas, como doação de código-fonte de importantes *softwares* (um exemplo recente foi o banco de dados Cloudscape) e colaboração ativa no aperfeiçoamento do *kernel* do Linux, com a criação de um laboratório espalhado por diversos países, inclusive no Brasil (em parceria com a Unicamp, em Campinas), com centenas de desenvolvedores escrevendo código que é doado à comunidade. A IBM também é uma patrocinadora ativa de diversas organizações que atuam nesse segmento, como a Apache Software Foundation e a OSDL (Open Source Development Laboratory), que é o ponto focal para o desenvolvimento do Linux e onde, inclusive, trabalha Linus Torvalds. Todos os computadores fabricados pela IBM rodam Linux, dos *desktops* aos *mainframes*, bem como todos seus principais produtos de *software* já foram portados para esse ambiente operacional.

Mas por que a IBM está tão intensamente envolvida com *software* livre e Linux? Para explicar este envolvimento vamos analisar o modelo de *software* livre e seu impacto na indústria de *software*.

Vejamos o exemplo do Linux. Este sistema operacional tornou-se rapidamente muito popular e é um exemplo de como um *software* livre pode ser bem-sucedido.

O Linux iniciou sua carreira na maioria das organizações como solução tática. Seus primeiros adeptos foram as famosas tribos, que romanticamente debatiam questões de *software* livre e grátis. Enquanto durou esse contexto, o ambiente empresarial ficou reativo. Implementar um *software*, sem um responsável final, não é um conceito facilmente aceito pelos executivos das grandes corporações.

Nos últimos anos, o Linux se profissionalizou, com concentração em poucos, mas sólidos distribuidores, e obteve comprometimento dos principais *players* do mercado de tecnologia da informação, como a IBM. Essas empresas influenciam o cenário dos *softwares* livres com seus parâmetros de disciplina, estabilidade e confiança, absolutamente necessários para que o sistema seja aceito no conservador ambiente corporativo, o que permite aos gestores de TI compreenderem que estão diante de um *software* avalizado pela comunidade empresarial.

Hoje o Linux ultrapassou o *tipping point* e é usado de maneira mais abrangente na maioria das empresas. O Linux já é uma idéia aceita pelos CIOs, que compreendem que, com esse ambiente, é possível obter níveis de escalabilidade, segurança e confiabilidade anteriormente apenas encontrados nos caros sistemas Unix/RISC, além de ser uma sólida alternativa ao Windows.

A questão mais atual passa então a ser o mundo pós-Linux. Os *softwares* livres como modelo vão se restringir ao Linux e a alguns outros poucos *softwares* como o Apache, OpenOffice, Postgres e MySQL, ou esta será uma onda irresistível que vai devastar toda a indústria de *software*?

Com certeza o movimento do *software* livre é de grande importância, pois afeta a estrutura de algumas das empresas mais sólidas e lucrativas da indústria de tecnologia. Obriga a revisão de práticas comerciais. Já vemos mudanças na própria maneira como o *software* passa a ser distribuído e remunerado. Ao invés de ser comprado em um pacote para ser instalado em seu computador, os usuários, sejam empresas ou indivíduos, podem agora usar o *software* e pagar de diversas maneiras. O *software* não precisa mais ser armazenado no computador da empresa, mas pode ficar hospedado em algum *site* remoto, que se responsabiliza pelas atualizações de versões e aplicações de *patches*, além de ser acordada uma taxa de aluguel para o seu uso.

Mas o novo contexto provocado pelo movimento do *software* livre afeta também de uma outra maneira as empresas do setor. A indústria de *software* não é homogênea em seus modelos de negócio. Existem diversos modelos de negócios, cada um com suas características e realidades.

Os *softwares* voltados para o mercado de massa, como suítes de escritório, têm custos incrementais de venda virtualmente zero, com estrutura de mercado altamente concentrada, modelo de receita baseado quase que exclusivamente em licenças de venda e manutenção. De maneira geral, pela alta

concentração de mercado, quase monopolista, tende a ser uma arquitetura proprietária e fechada.

Os impactos causados pelo modelo de *software* livre podem causar sérios transtornos exatamente nas empresas extremamente dependentes de receitas advindas quase que exclusivamente de licenciamentos de aplicativos em seus modelos de negócio. Um novo cenário, causado por uma tecnologia de ruptura que atua diretamente na faixa de mercado da empresa, torna o seu crescimento mais árduo, gerando por parte dela reações bastante agressivas para evitar eventual percepção negativa dos seus investidores.

Por outro lado, as empresas que atuam intensamente em serviços e obtêm um percentual relativamente menor de sua receita total em *software*, podem rapidamente aumentar sua receita de serviços entrando fortemente nesse novo segmento.

Portanto, entendemos que o modelo de *software* livre vai transformar a indústria de TI, mas não de maneira uniforme. Para a IBM, empresa que está se transformando rapidamente em uma fornecedora de soluções de negócios, embasada por forte domínio da tecnologia de suporte, o modelo de *software* livre permite que suas iniciativas em inovação se concentrem nas camadas de aplicações de maior valor agregado, atuando em sinergia com a comunidade no desenvolvimento de sistemas operacionais e funções básicas de banco de dados.

Conviveremos sob ambos os modelos, *softwares* livres e proprietários, e caberá aos usuários e à indústria extrair o melhor das alternativas disponíveis. ■

\* Cezar Taurion é gerente de novas tecnologias aplicadas da IBM Brasil

recebido visitas, feito reuniões, palestras, enfim, temos procurado ajudar outras empresas interessadas em reduzir seus custos através de medidas como essa".

No início de 2002, a Sun lançou a versão 6.0 do StarOffice, fechando o código e passando a cobrar a licença, mas em paralelo criou um outro produto com código aberto e todos os componentes não proprietários do StarOffice 5.2, o OpenOffice.org, "o maior projeto de código aberto do mundo". O Metrô-SP adotou o OpenOffice.org como sua ferramenta padrão de suporte às atividades de escritório. "O apoio da comunidade brasileira de *software* livre é tão forte – constata Gustavo Mazzariol – que apenas alguns meses depois da primeira versão do OpenOffice foi lançada a versão traduzida para português – Brasil, bem antes da Sun lançar sua tradução do StarOffice 6.0. E em pouco tempo incrementaram no OpenOffice as melhorias que existiam no StarOffice 6.0."

### Situação atual

Hoje praticamente todos os equipamentos do Metrô-SP utilizam o OpenOffice.org. Para as estações mais antigas, menos robustas, no entanto, são adotados o Word 97 e o Excel 97. Os novos micros têm o OpenOffice e à medida que são trocados deixam de usar o Word/Excel. Os micros mais novos não estão autorizados a utilizar as licenças existentes do Word/Excel-97.

O Metrô-SP desenvolve ainda outro projeto, o Micro Livre, que disponibiliza para as áreas novos micros contendo apenas *software* livre e gratuito. "Já a utilização do Linux em nossos servidores é corriqueira. Temos ampliado gradativamente esse uso. Desde nossas experiências iniciais com o correio, todo o gerenciamento da nossa rede é também realizado com *software* livre. Mas não nos fixamos em uma única distribuição Linux."

### Vantagens

Uma das grandes vantagens indiretas apontadas por Gustavo Mazzariol com a adoção do *software* livre

foi a mudança de cultura, de posturas, e um novo incentivo para a busca por outras alternativas, sejam livres, gratuitas ou mais baratas. "Trabalhamos com bancos de dados livres, CAD genérico e utilitários gratuitos. Hoje temos aliados entre os nossos usuários que assumiram também tais posturas e vivem nos passando dicas de produtos, de alternativas. Não ficamos mais à mercê de grandes empresas de *software*. Continuamos a adquirir licenças de *software*, quando necessário, pois não somos radicais a ponto de não reconhecer e usar *software* proprietário onde cabível e recomendável."

Outro benefício foi o ingresso da empresa numa comunidade que usa e defende o código aberto. "Conhecer tantas pessoas que simplesmente ajudam umas às outras, sem outro interesse que não seja colaboração, é surpreendente. E mais surpreendente ainda é assistirmos o verdadeiro *boom* de *software* livre que acontece no Brasil e no mundo, principalmente nos dois últimos anos."

Atualmente os produtos livres são utilizados por todas as áreas da empresa, em menor ou maior escala. "Temos feito grandes economias, mesmo considerando o que gastamos com estudos, treinamentos e implantações. Normalmente, só as economias obtidas nos custos das licenças já compensam. Temos aproximadamente 2.050 micros e em todos eles são encontrados produtos livres ou gratuitos. Dos 7.300 empregados, quase 5.000 são usuários constantes de *software* livre."

O balanço da adoção do *software* livre é positivo: "descobrimos que estamos no caminho certo. Usar *software* livre foi uma decisão estratégica, ousada e inovadora. Hoje temos certeza de que acertamos e se continuarmos nessa linha estaremos sempre progredindo. Se no começo andávamos sozinhos e contra o vento, hoje temos inúmeros parceiros andando à frente, ao lado e nos seguindo. O rumo de todos é o mesmo: ao sucesso."

### A economia

Segundo Gustavo Mazzariol, a economia obtida é de algo em torno de R\$ 3 milhões/ano, no seguinte cenário:

- uso do OpenOffice.org e conseqüente economia de licenças do MS-Office, para um parque de 2.000 micros, a US\$ 300 por micro por ano;
- uso de bancos de dados livre como o PostgreSQL e o MySQL, economizando vários servidores Oracle que têm um custo de US\$ 45 mil por processador;
- uso do MetroMail, correio todo baseado em *software* livre. Um correio proprietário equivalente custa em torno de R\$ 100 por usuário por mês. São 5 mil contas e em breve estaremos expandindo para perto de 8 mil contas;
- nossa intranet é toda baseada em *software* livre;
- gerenciamento de rede usando *software* livre;
- uso de Linux em vários servidores, economizando em licenças de Windows Server (NT/2000/2003).

# EMPREL

## EMPRESA MUNICIPAL DE INFORMÁTICA

### A implantação

O uso de *software* livre na Emprtel – Empresa Municipal de Informática da Prefeitura de Recife – foi adotado a partir de março de 1997. Segundo o presidente da entidade, Waldemar Borges, tudo começou com a substituição de um servidor RISC, que apresentou problemas. "A estimativa de custo para solução do problema era de R\$ 4.500 para substituição do disco rígido e R\$ 8.500 para *upgrade* do sistema operacional." Optaram, então, pela contratação de serviços de consultoria para implantação do Linux como servidor de *e-mail*, DNS e autenticação, que custou R\$ 600; a aquisição de um servidor Intel custou mais R\$ 2.500. Essa mudança foi feita por um consultor independente, contratado para instalação da primeira versão do Linux, e os custos, absorvidos sem problemas.

"A substituição do servidor ocorreu em uma semana sem maiores dificuldades técnicas", lembra Borges. Para a introdução dos demais *softwares* livres, existia o receio natural da mudança, que foi sendo superado com os bons resultados das sucessivas experiências, além das palestras e discussões sobre o tema. A equipe envolvida era composta apenas pelo consultor e mais dois técnicos. Não houve nesse momento um treinamento específico para os técnicos da Emprtel. Eles utilizaram a documentação disponível na internet para o auto-aprendizado e orientações do consultor. Para prosseguimento do projeto, no entanto, com a introdução dos demais *softwares* livres, a Emprtel treinou multiplicadores, responsáveis por novos treinamentos para funcionários da Prefeitura e para a população em geral.

Com relação à orientação legal, Waldemar lembra que no início não houve qualquer suporte para essa implantação. Segundo ele, a Lei 16.639/2001, de sua autoria, que prioriza o uso do *software* livre na Prefeitura, só foi sancionada pelo prefeito João Paulo em 2001. Posteriormente, em 2003, a Emprtel instituiu, através de portaria, o Padrão Tecnológico de Referência, que indica a utilização de um conjunto de *softwares* livres para a Prefeitura.

A estratégia utilizada foi a de implantar primeiramente o Linux, os *softwares* servidores e o ambiente de desenvolvimento, para posteriormente investir na implantação de *softwares* de produtividade pessoal e de colaboração. A introdução dos demais *softwares* aconteceu de forma lenta e gradual, com seus custos sendo, da mesma forma, absorvidos.

Em 2004, a Emprtel lançou o Portal de Software Livre do Recife (<http://sl.recife.pe.gov.br/>) e criou, em conjunto com outras instituições públicas federais e estaduais e universidades, a CSLPE – Comunidade de Software Livre de Pernambuco

Divulgação



"Estimamos uma economia de R\$ 1,9 milhão até o momento, somente considerando o uso do Linux e do OpenOffice".

(<http://www.softwarelivre.pe.gov.br/>). Atualmente está avaliando a adoção de *softwares* livres para o geoprocessamento e desenvolvendo um *software* livre chamado SIRI - Sistema Integrado de Gestão de Recursos da Internet (<http://sl.recife.pe.gov.br/projects/siri/>).

No entanto, explica o presidente da Emprel, o avanço mais rápido do projeto exige investimentos específicos. "Uma parte desses recursos está vindo do PNAFM – Programa Nacional de Apoio à Gestão Administrativa e Fiscal dos Municípios Brasileiros, com recursos do BID – Banco Interamericano de Desenvolvimento – e com a Caixa Econômica Federal como órgão operador do programa no Brasil." Adicionalmente, a Prefeitura recebeu equipamentos, numa parceria com o Banco do Brasil, e montou cinco laboratórios com *softwares* livres em escolas públicas. Hoje a Prefeitura utiliza 52 *softwares* livres, desde o sistema operacional, passando por *firewalls*, servidores de *e-mail*, de DNS, de banco de dados, de aplicações, automação de escritório, *softwares* para o ambiente de desenvolvimento, etc.

## A situação atual

Atualmente o município tem 300 funcionários da Emprel e outros mil funcionários da Prefeitura utilizando um total de 1.300 máquinas com *softwares* livres. A Prefeitura do Recife utiliza *softwares* livres em 1.272 máquinas *desktop* – de um total de 5.000 máquinas – e em 28 das 34 máquinas servidoras. As máquinas *desktop* utilizam o OpenOffice, muitas com o Linux; e as máquinas servidoras utilizam, no mínimo, o Linux. Já a Emprel utiliza *softwares* livres em 90% de suas máquinas e o restante encontra-se na administração direta e indireta da Prefeitura.

Waldemar Borges destaca a Secretaria de Educação, que adota o OpenOffice nas UTECs - Unidades de Tecnologia na Educação e Cidadania, em laboratórios de escolas públicas e nos ônibus-escolas, oferecendo treinamentos e acesso à internet gratuitos para a população. "Estimamos uma economia de R\$ 1,9 milhão até o momento, somente considerando o uso do Linux e do OpenOffice", contabiliza.

## Recomendações

Com a experiência adquirida na Emprel, Waldemar Borges aconselha: é importante que a empresa que se propõe à adoção do *software* livre crie um clima favorável para a mudança. Isso pode ser construído através de palestras, apresentação de casos de sucesso e informações objetivas sobre os *softwares* livres: razões para sua adoção, benefícios e dificuldades esperadas e como superá-las. "Recomendamos um plano de implantação", sugere. "Um bom subsídio é o Guia Livre, produzido pelo Governo Federal (<http://www.softwarelivre.gov.br/>). Entre outras ações, o plano deverá contemplar a proteção aos dados atuais, o treinamento e a garantia do suporte adequado aos usuários, a fim de evitar que as dificuldades encontradas no dia-a-dia sejam motivo para reversão do processo de mudança. E, finalmente, é preciso compreender que *software* livre não significa *software* de graça, e, portanto, o projeto deverá ser orçado e ter o investimento garantido."

## O futuro imediato

- Todas as estações de trabalho da Emprel com Linux e OpenOffice até o final de 2005;
- Todas as estações de trabalho da Prefeitura do Recife com Linux e OpenOffice até o final de 2006;
- Implantação dos sistemas desenvolvidos dentro do PNAFM até o final de 2006;
- Aprimorar o SIRI - Sistema Integrado de Gestão de Recursos da Internet ;
- Treinamentos para a Prefeitura e para a população do Recife;
- Implantação de geoprocessamento baseado em *softwares* livres;
- Participação ativa da Emprel na comunidade, em especial na CSLPE - Comunidade de Software Livre de Pernambuco;
- Participação ativa da Emprel na equipe de coordenação e nas palestras, dentro do LACFREE 2005 - Conferência Latino-americana e do Caribe para Uso e Desenvolvimento de Software Livre, que será realizada em outubro de 2005 em Recife/Olinda;
- Estímulo ao uso de *software* livre no mercado local de informática (Porto Digital);
- Criação de Praças da Informação, ou seja, novos centros de treinamento e acesso à internet para uso gratuito pela população de Recife.



O desafio do presidente da Celepar, Marcos Vinícius Mazzoni, quando assumiu a entidade há aproximadamente dois anos e meio, ultrapassava a adoção de uma tecnologia específica. Segundo ele, a missão era resgatar a empresa de informática como centro de tecnologia do Governo, revertendo a situação existente de um modelo de terceirização de serviços. "Tudo o que era novo estava sendo feito por empresas privadas terceirizadas; os grandes projetos estavam em processo de licitação ou já tinham sido licitados", lembra.

O objetivo foi fortalecer novamente a Celepar que, segundo Mazzoni, apresentava um parque de máquinas defasado, fazia manutenções em sistemas que ela sequer havia desenvolvido ou em seus sistemas legados tradicionais. "Tivemos que investir no parque de máquinas, substituir servidores, trocar o *mainframe* por novas tecnologias, inclusive com ambientes Linux. Enfim, substituir toda a plataforma de desenvolvimento para obter mais produtividade com o *software* livre; fizemos também a migração de todos os serviços que estavam fora, terceirizados, como era o caso do Detran, da Saúde, da Educação. Tudo voltou para a Celepar."

Para isso, foi necessária a requalificação de todo o quadro técnico da empresa e a contratação de 180 profissionais, por concurso público, só para a área de desenvolvimento. Há previsão de novo concurso ainda este ano para recompor a equipe.

Para mudar todo o conceito de desenvolvimento de forma rápida, foi criada uma sistemática de cooperação com a comunidade de desenvolvedores. "Isso evidentemente só é possível no mundo do *software* livre", constata o presidente da Celepar. "Nós ajudamos a constituir o Movimento do Software Livre do Paraná e, junto com a comunidade, começamos o treinamento dos nossos profissionais, a estudar migrações, a ver como é que nós poderíamos alterar as coisas até chegar ao que nós chegamos agora, nesse mês de abril: uma nova plataforma completa de desenvolvimento de *software* livre, em que todos os nossos desenvolvedores usam máquinas com Linux e têm num servidor todas nossas ferramentas de desenvolvimento de *software* livre".

---

"Economizamos, em licença de *software* que deixamos de comprar, em torno de R\$ 40 milhões nesses dois anos".

---

Para isso foi adotado o conceito de fábrica, em que quarenta profissionais só desenvolvem: não fazem manutenção, não atuam nos sistemas legados e trabalham exclusivamente com ferramentas de *software* livre. O balanço é o seguinte: o que antes era 100% plataforma proprietária é hoje 80% servidores Linux; das 900 máquinas de trabalho, 800 já estão totalmente em Linux; já foram migradas várias secretarias e criado o projeto da Educação. Nessa área, são hoje 18 mil máquinas com Linux e a previsão é de que a partir do segundo semestre esse número chegue a 36 mil e em 2006, a 56 mil máquinas, representando a informatização das 2.200 escolas integralmente em *software* livre.

De acordo com Mazzoni, o orgulho da equipe da Celepar é o portal [www.diaadiaeducacao.pr.gov.br](http://www.diaadiaeducacao.pr.gov.br), criado para a rede escolar, com o objetivo principal de apoiar os professores. Obedecendo a mesma premissa de compartilhamento e colaboração do *software* livre, os professores têm acesso a conteúdos dos diversos campos do conhecimento disponibilizados por colegas integrantes da rede. "A Secretaria colocou todos os conteúdos de Matemática, de História, de Geografia, nesse portal; é um dos nossos projetos mais importantes", comemora Mazzoni. No ambiente do portal os professores entram e discutem as matérias por meio de um *chat*; todos os professores da rede pública estadual podem colaborar e o portal está aberto para o Brasil inteiro.

Outro produto da empresa é o Correio Expresso, desenvolvido a partir de uma ferramenta da comunidade alemã. São atualmente mais de 30 mil usuários. "Ele é todo em *software* livre e vai ser o correio da rede dos alunos, ou seja, nós vamos ter mais de 1 milhão e meio de usuários nesse correio. É voltado para a língua portuguesa, todo ele integrado com agenda, correio e catálogo; é uma ferramenta de *groupware* completa. Temos hoje em torno de 40 projetos em andamento no mundo do *software* livre", informa Mazzoni.

Os resultados já podem ser contabilizados: "Economizamos em termos de licença de *software* que deixamos de comprar em torno de R\$ 40 milhões nesses dois anos. Além disso, há os projetos já concebidos em *software* livre, em que são adquiridas máquinas sem *software*. Quem coloca o *software* somos nós".

Os 36 telecentros existentes hoje no Estado do Paraná também estão todos em *software* livre. O projeto prevê 200 telecentros com *software* livre, que serão integrados com a escola.

"Através do *software* livre nós viabilizamos inclusive o desenvolvimento de todos os projetos finalísticos do Governo e conseguimos manter nosso pessoal com domínio total da tecnologia. Dessa forma, mesmo o que temos que comprar fora compramos com base em nossas especificações, porque qualquer produto para ser vendido para o estado do Paraná tem que ser totalmente aderente ao *software* livre. Isso garante a continuidade dos projetos pelos nossos profissionais."

Esse momento é especialmente importante para o Estado, enfatiza Mazzoni, "Porque fechamos as ferramentas de desenvolvimento e temos agora a estrutura para os desenvolvedores, que vão gerar exclusivamente *software* livre." Ele lembra que a experiência de implantação do *software* livre no Rio Grande do Sul - em 1999, quando era presidente da Procergs - foi importante porque ajudou a quebrar resistências. "A experiência anterior nos ajudou a passar, agora, por cima da discussão da tecnologia e nos concentrar no posicionamento estratégico do governo. E se foi possível fazer tudo que se fez com *software* livre no Rio Grande do Sul, era possível fazer em outro lugar."

Se por um lado havia no Paraná um ambiente mais receptivo à mudança, por outro havia o problema da transferência de todos os contratos terceirizados para dentro da Celepar, tecnologias que a empresa não escolheu ou "sequer teve chance de opinar em gestão anterior" e migrar tudo isso para soluções em tecnologia de *software* livre, a custos inferiores aos contratados. "Foi nosso desafio inicial para provar o modelo: não podíamos deixar mais caro, tínhamos que conseguir fazer com que tudo isso acontecesse com custos menores." Como exemplo, ele lembra o sistema da área de Saúde, que era terceirizado e custava ao Estado R\$ 750 mil reais por mês; ao assumir o sistema, a Celepar fez a migração em 20 dias e passou a cobrar R\$ 50 mil reais por mês.

# Software Livre na Procergs



Carlos Alberto Pacheco\*

**S**oftware Livre não é novidade na Procergs. Desde 1995, por ocasião do lançamento pela Procergs do primeiro provedor internet no Rio Grande do Sul, o Via RS, o mesmo foi viabilizado pela utilização majoritária de *software* livre na composição de sua infra-estrutura.

O posicionamento da atual gestão da Procergs em relação ao *software* livre sempre foi muito claro e objetivo. Desde o primeiro momento que assumimos a companhia, dedicamos muito tempo em estudos e avaliações em relação a esse tipo de tecnologia, por entender que se tratava de uma realidade de mercado irreversível. Adotamos, porém, uma visão mais pragmática em relação ao tema, descartando qualquer tipo de postura ideológica e tendenciosa. Entendemos que tanto o *software* livre como o *software* proprietário são apenas ferramentas e como tal deve ser levado em conta o custo/benefício no momento da opção.

Seguindo essa linha de pensamento, entendemos que a aplicação de *software* livre na administração pública estadual deve ser analisada com muito cuidado e responsabilidade, exigindo um estudo "caso a caso".

É consenso hoje em dia o fato de que não existem soluções em *software* livre para todas as necessidades técnicas. Além disso, no que tange à distribuição deste, existem muitas dúvidas em relação ao respaldo jurídico para se fazer isso. Afinal, pode a administração pública disponibilizar algo de forma gratuita à sociedade sem ser punida legalmente? Esta é uma questão que mereceria uma discussão mais profunda por parte dos gestores que vêm adotando o *software* livre em suas soluções.

Na Procergs, está implementado um modelo de negócios em relação ao *software* livre que se baseia na venda de soluções, postura esta aderente ao próprio negócio da empresa, qual seja: "soluções em tecnologia da informação e comunicações."

Um exemplo prático aqui na empresa é o caso do correio corporativo integrado com agenda e catálogo de endereços com acesso pela internet, que desenvolvemos com base no *software* livre. O Direto, como o chamamos, é um aplicativo que, antes de assumirmos a direção da empresa, vinha sendo distribuído sem os necessários cuidados em relação ao regramento jurídico e legal. Várias empresas estaduais adotaram o Direto como solução de correio eletrônico na oferta de serviços às administrações estaduais. Dentre os regramentos que consideramos significativos estão a preservação do produto como *software* livre com um novo modelo de licenciamento para novas versões, e a obtenção do controle sobre a evolução tecnológica dos produtos desenvolvidos. Procuramos preservar as marcas de propriedade da Procergs, estabelecer um modelo que preserve a autoria dos códigos desenvolvidos pela companhia e também por outrem, permitir e viabilizar a coexistência de *software* livre com *softwares* proprietários e acima de tudo o respeito pela legislação vigente. ■

\* Carlos Alberto Pacheco de Campos é diretor-presidente da Procergs



# Conhecimento gera Conhecimento

Paulo César Lopes\*

Navegar é preciso, viver não é preciso...

(Fernando Pessoa)

**H**oje o conhecimento é motor que gera riquezas e supera gradativamente os tradicionais fatores de produção: a terra, o trabalho e o capital. A maioria das empresas, atentas às mudanças e ao mercado, deixou de produzir para realizar serviços e vender conhecimento, agregando valor a seus produtos. Segundo o Institute of the Future<sup>1</sup>, o setor de serviços é o que mais cresce na economia mundial e as empresas e negócios baseados no uso intensivo do conhecimento são o futuro. Hoje, mais de 45% do PIB dos países desenvolvidos resultam dessa força motora.

Trata-se de um recurso inovador que, ao contrário da terra, do trabalho e do capital, não se exaure com a sua utilização e não apresenta limitações físicas. Mais: conhecimento gera conhecimento. É o imaterial gerando o imaterial, a exemplo do que ocorre no mundo digital. Isso traz novas perspectivas para a economia e a sociedade, determinando uma maior complexidade nas relações entre os segmentos sociais e uma preocupação constante com a gestão do conhecimento.

O principal motivador dessa complexidade é o processo de *autopoiese*<sup>2</sup>. O conhecimento surge do conhecer e o conhecer gera conhecimento. As implicações filosóficas, sociológicas e econômicas desse processo são imensas e foram catalisadas com o advento da grande rede. Na internet você

obtem desde o endereço de um indivíduo até a receita para se confeccionar uma bomba atômica. Mas bem antes do surgimento da grande rede o conhecimento livre e sem restrições já existia<sup>3</sup>. Se a internet possui hoje dezenas de milhões de *sites*<sup>4</sup>, vários são mantidos pela força da colaboração e do cooperativismo, com a aplicação dos novos princípios adotados pelo desenvolvimento técnico e científico moderno: "hospitalidade"; "reciprocidade"; "solidariedade" ou "colaboração solidária"; e o princípio do "autodesenvolvimento e estima."

O primeiro deles permite a aceitação do outro, ou dos outros, da maneira que se apresentam. O segundo princípio considera a possibilidade do escambo de propostas, teorias, ferramentas, metodologias e conhecimentos. O terceiro é bem claro e busca o trabalho cooperativo e solidário entre as pessoas e grupos de pessoas. O último princípio considera o colaborador como parte importante de um todo muito maior que suas colaborações, ensejando o "trabalho com paixão" e autodesenvolvimento.

Tais princípios são revolucionários e estão presentes no movimento pelo conhecimento livre. São as bases das mudanças de paradigmas propostas e estão baseadas no ambiente acadêmico. Toda tese acadêmica desenvolvida por um único pesquisador, ou por um grupo deles, deve ser apresentada

1 [http://www.iftf.org/docs/2003\\_Ten-Year\\_Forecast.pdf](http://www.iftf.org/docs/2003_Ten-Year_Forecast.pdf)

2 Autopoiese quer dizer autoprodução e autogeração. A palavra surgiu pela primeira vez num artigo publicado por Varela, Maturana e Uribe, para definir os seres vivos como sistemas que produzem continuamente a si mesmos (1974).

3 [http://www.suigeneris.pro.br/edvariedade\\_jgarcia.htm](http://www.suigeneris.pro.br/edvariedade_jgarcia.htm)

4 <http://news.netcraft.com/>

para a comunidade. Nessa apresentação, aberta ao público, todo o conhecimento obtido e gerado pela pesquisa deve ser colocado de maneira transparente para a comunidade, incluindo sua publicação em uma revista especializada. Essa é a cultura que foi construída por centenas de anos para o conhecimento acadêmico teórico e prático. A comunidade de especialistas, por sua vez, deverá testar, avaliar, criticar e, caso haja problemas, refutá-la. Esse processo obriga a exposição das idéias, ferramentas, processos e de todo o conhecimento envolvido. Novamente, o conhecimento gerando conhecimento.

Evidentemente que existem exceções. Muitos centros de pesquisas, distantes do mundo acadêmico, não seguem essa metodologia. As justificativas mencionam questões de segurança e razões financeiras ou estratégicas para a manutenção dos conhecimentos como segredos do negócio. O que coloca em um extremo o paradigma de manutenção do conhecimento proprietário, que considera o saber como objeto de propriedade individual ou empresarial, e no outro extremo o paradigma revolucionário do conhecimento livre, que considera o saber como patrimônio da humanidade. Discussão polêmica que esbarra naturalmente na clássica questão da propriedade e seus efeitos sociais. Tema complexo que já produziu várias teses e trabalhos, mas que sempre permite discussões acaloradas e bastante irracionais. Quando se presencia tais conflitos, facilmente identificam-se as dificuldades de comunicação entre os defensores de ambos os paradigmas. A semântica e a carga léxica de ambos os lados não permite uma comunicação sem ruídos. Emissor e receptor sempre estão interpretando as informações de maneira tendenciosa.

Mas algumas conclusões são perceptíveis. Uma delas é a consolidação dos novos princípios já referidos neste artigo. A grande rede os expôs ao

mundo<sup>5</sup>, transformando o trabalho, que pode ser bastante cooperativo e solidário na rede. A solidariedade e hospitalidade ganharam outra dimensão e uma interpretação mais ampla. Não há como um usuário não se sentir como parte da rede.

A internet catalisou todo o processo de exposição e construção do conhecimento e da elaboração de novas teses e pesquisas. Abriu a possibilidade da "navegação" por todo tipo de conhecimento. "Navegar" tornou-se uma necessidade básica para a aprendizagem e para a busca e produção do conhecimento e incentivo à participação de todos na elaboração dessa nova dinâmica. Quase todo o usuário da internet já esbarrou no conhecimento livre, quase sempre por meio de pesquisas nos sites de busca. Mas o que poucos percebem é que a maioria das pesquisas apresenta resultados obtidos do maior projeto de compartilhamento de conhecimentos na rede: a Wikipedia<sup>6</sup>. Essa enciclopédia é o resultado do trabalho de colaboração de milhares de milhares de indivíduos, oferecendo hoje mais de 500.000 verbetes em língua inglesa, 220.000 em alemão, 100.000 em japonês e dezenas de milhares em francês, português, espanhol, polonês, etc. (são mais de 100 línguas e 1.5 milhões de artigos/verbetes). Tornou-se a maior enciclopédia disponível no planeta em número de verbetes. Novos verbetes, correções, ou complementações podem ser propostos por qualquer um. Os artigos/verbetes devem ser neutros e são avaliados por um comitê gestor e pela própria comunidade. A participação dos internautas é espantosa e demonstra bem o poder da solidariedade e da reciprocidade dentro da rede. É a força da antiga proposta do conhecimento livre. ■

\* Paulo César Lopes é bacharel em Computação, Administração e Direito (UFMG), pós-graduado em Computação (UFMG) e analista de suporte da Prodemge

5 A internet também propiciou o surgimento de propostas de novas relações sociais, novas relações de negócios e de uma nova ética entre as pessoas e as empresas. Um exemplo claro é o Manifesto Cluetrain, ou o Trem das Evidências, que surgiu em meados de 1999, e tornou-se rapidamente um baluarte. Hoje, encontram-se no seu site mais de 200 páginas de assinaturas. Nelas várias pessoas influentes de vários segmentos do conhecimento humano assinam e tecem comentários. <http://www.cluetrain.com/portuguese/>

6 <http://wikipedia.org>



## Racionalização de Recursos em TI Usando Softwares Livres e Outras Alternativas

### Helvécio Borges Filho

Engenheiro aeronáutico, formado e com mestrado pela Universidade de Illinois nos Estados Unidos. Trabalhou na Embraer como projetista de aeronaves e nos EUA como consultor na área de simuladores de voo.

É instrutor autorizado do banco de dados livre MySQL e sócio-diretor da EAC Software, especializada em desenvolvimento de soluções que utilizam tecnologias livres e que rodam em qualquer plataforma. É vice-presidente da Assespro-MG (entidade representativa do meio empresarial de informática).

### RESUMO

Este artigo analisa a situação atual da TI no Brasil no que tange ao uso dos principais *softwares* adotados pelo mercado (*mainstream*) e o potencial de uso de *softwares* alternativos e livres, de custo inferior. São citadas situações onde se pode racionalizar recursos com produtos de menor custo, assim como situações não favoráveis ao uso de produtos livres. O objetivo é dar a devida atenção a todas as opções que o mercado de TI oferece, às necessidades dos clientes finais e às várias considerações que compõem o quadro de decisão, procurando valorizar mais, em última análise, os interesses de longo prazo das organizações e o papel dos gestores de TI.

Existem no Brasil, mais do que em outros países, vários mitos em torno do *software* livre. Como engenheiro e empresário de TI, vejo *softwares* livres e outros *softwares* proprietários fora do *mainstream* como alternativas tecnológicas interessantes, merecedoras de estudo e avaliação, para serem utilizados em situações em que se mostrarem mais vantajosos, por diversos

critérios, que variam para cada projeto.

Nosso desconhecimento do assunto, por uma série de motivos, tem criado dificuldades na adoção de soluções alternativas, com potencial de gerar grandes economias e/ou domínio tecnológico estratégico para as organizações. E, enquanto países ricos do primeiro mundo usam simultaneamente todas as opções tecnológicas

que o mercado oferece, ficamos aqui no Brasil nos debatendo com questões de segunda ordem, como preferências pessoais, com radicalismos de todos os lados, e perdemos o foco do negócio, que é gerir a TI da forma mais eficiente. Com isso estamos perdendo a preciosa vantagem tecnológica que detínhamos na década de 70 e 80. Se continuarmos assim, ficaremos relegados a



arrastadores de componentes, a clicadores de mouse, reféns de tecnologias que não dominamos. Será que é este nosso destino, nossa vocação como país?

Ademais, a busca por padrões abertos tem sido uma antiga e justa reivindicação nos projetos de governo eletrônico (*e-gov*), atendendo simultaneamente o Governo e, não com menos importância, o cidadão, cliente e pagador dos serviços. Entretanto, esses projetos de *e-gov* têm, via de regra, estimulado a pirataria na medida em que forçam o cidadão a trocar com o Governo informações em formatos proprietários (editais, *sites*, bancos e até recentemente a declaração de imposto de renda).

A troca de informações através de protocolos padronizados e conhecidos (padrões abertos e interoperáveis) é algo de livre utilização e fácil implementação em quase todas as situações, não sendo mais necessária a adoção de soluções padronizadas de protocolos fechados (ou proprietários, as chamadas "caixas-pretas").

Hoje em dia:

- qualquer sistema operacional gera arquivo no formato PDF para ser lido em qualquer outro sistema operacional;
- qualquer aplicação, escrita em qualquer linguagem, rodando em qualquer sistema operacional, "conversa" com qualquer banco de dados rodando em qualquer outra plataforma;

- escrevem-se *e-mails* que podem ser lidos em qualquer sistema operacional ou programa;
- um *site* pode ser desenvolvido de forma que rode em qualquer plataforma e com qualquer navegador compatível com padrões internacionais.

Com um pouco de atenção aos clientes dos serviços, aos recursos tecnológicos de que dispomos e com boas diretrizes de desenvolvimento e testes, podemos facilmente chegar a um ótimo patamar de interoperabilidade.

Ou seja, o que não faltam são formas de fazer com que produtos diferentes conversem entre si, otimizando os recursos das organizações, da mesma forma que um administrador de uma frota de veículos racionaliza seus recursos utilizando produtos diferentes para aplicações diferentes.

Sob essa ótica, a homogeneidade de soluções pode, em muitos casos, ser contraproducente. E, mesmo sendo contraproducente para um órgão governamental, qualquer projeto de governo eletrônico tem que atender também ao cidadão, na plataforma em que ele estiver, sob pena desse projeto se tornar mais um forte estímulo à pirataria que o Governo tanto tenta combater.

No passado, havia a impressão de que, no caso de adoção de alguma outra solução fora do *mainstream*, seria necessário um treinamento especial para todos os usuários. Com o advento da

interface gráfica, a interação entre o usuário e a máquina é totalmente intuitiva, materializada através de ícones e *helps* de toda espécie e em todas as línguas. Para salvar um documento, o ícone é um disquete em qualquer plataforma. Para imprimir, clica-se no ícone de uma impressora e assim por diante. As funcionalidades dos produtos que vingaram no mercado são próximas demais para se justificar a adoção das soluções mais caras, exceto em situações muito bem justificadas.

A melhor forma de preservação do investimento em capacitação é treinar o usuário a utilizar informática, de forma genérica, e não em um produto em especial. Ninguém adquire carteira de habilitação para dirigir um carro de uma só marca, apesar das pequenas diferenças em carros de marcas diferentes. Quem usa um editor de texto, usa qualquer editor de texto de qualquer fabricante, exceto quando o usuário utiliza recursos muito avançados e necessita de algum curso ou suporte especial. A dificuldade de transição entre aplicativos para atender o cidadão ou trabalhador comum reside apenas no plano psicológico.

De toda forma, é fato que a grande massa de usuários de informática no Brasil, independentemente de estarem utilizando em casa ou em ambiente corporativo, adota *softwares* proprietários. Mas vale a pena entender um pouco o ambiente que nos levou a esta condição.



## A História do Software Proprietário Livre no Brasil

Os primeiros *softwares* livres no Brasil foram proprietários, senão vejamos.

Com a facilidade que a pirataria proporcionou no final dos anos 80 até o presente momento, os *softwares* proprietários se tornaram praticamente livres, com livre circulação, com livre duplicação, e os usuários e gestores fecharam os olhos para as questões de licenciamento, custo, critérios de instalação, funcionalidade, segurança, privacidade e pensamento de longo prazo.

O alto preço dos produtos importados, mesmo os mais simples, impingia uma condição que impedia e impede até hoje que, em sua consciência, um cidadão ou empresário adquira esses produtos legalmente e de forma indiscriminada como ocorre atualmente na grande maioria das situações. Não há dinheiro no Brasil para pagar todo o *software* proprietário instalado e nem haveria cadeia para abrigar tanto contraventor.

Impulsionados também pela inquestionável qualidade da quase totalidade dos produtos importados de TI, criou-se no Brasil uma impressionante massa crítica de usuários piratas.

Na ausência de preço, o usuário, com o advento do microcomputador, tornou-se dono e gestor soberano do seu equipamento e passou a escolher o que melhor lhe convinha, por qualquer critério, seja técnico, seja por influência de colegas ou pela mídia (poder econômico, com alto poder de influência tanto nos decisores quanto nos usuários). O fato é que ele, usuário ou gestor, se sentiu livre, livre de ter que pagar, livre para escolher.

Com o passar do tempo, estes usuários piratas se tornaram também competentes nos produtos que eles livremente instalaram.

E, já que é "livre", não de direito mas de fato, não havia mais, como outrora, necessidade de se questionar a qualidade de alguns destes produtos, ou mesmo testar e homologar. "Gratuito", "interoperável", por que não continuar nesta linha?

Este cenário, aliado ao alto custo mesmo dos aplicativos mais simples, criou no Brasil um ambiente favorável à homogeneidade de aplicativos proprietários, notadamente aqueles fabricados pela Microsoft, mas não restritos a estes de forma alguma.

As campanhas anti-pirataria, quase sempre tímidas e financiadas pela indústria estrangeira de *software* proprietário, obviamente nunca se preocuparam muito em orientar o cidadão sobre alternativas mais econômicas, que sempre existiram, pagas ou gratuitas, proprietárias ou livres.

## Por Que Usar Softwares Livres e Outros Softwares Alternativos?

São muitas as possíveis razões para que um gestor de TI, analista ou programador, se veja atraído, cogite ou seja incentivado, em diversos graus, a adotar *softwares* livres ou outras alternativas.

É inegável a economia que algumas situações proporcionam. Um exemplo simples de uma situação dessas é a escolha de um aplicativo para escrever textos, planilhas ou fazer apresentação de *slides*. Não faz nenhum sentido onerar uma administração com

*softwares* pagos nessa categoria, exceto em situações muito especiais e bem justificadas. O WordPad, editor de texto gratuito no Windows, atende a 90% dos atuais usuários do Word, sendo que este custa 1.600 reais. O WordPerfect custa 300 reais. O

OpenOffice é gratuito. Para quem não está pagando a conta (ou acha que não está), o melhor produto é aquele que o próprio usuário julga melhor e não o que convém à organização para a qual ele trabalha.

Há ainda condições em que o gestor financeiro, o gestor



jurídico ou o gestor estratégico podem também vir a atuar de forma preponderante sobre as decisões na área de TI, seja por questões de custo, por questões de legalidade, privacidade ou segurança, de forma incondicional, não importando nestes casos a indicação técnica ou de mercado (seguir o que o usuário deseja).

A Nasa não usa *software* proprietário em seus veículos espaciais por questões de segurança.

Todos os novos filmes que utilizam efeitos especiais lançam mão de *softwares* livres customizados porque isso lhes dá vantagem competitiva, um diferen-

cial que não seria possível se usassem apenas os recursos dos *softwares* proprietários. Quando se usa um *software* "caixa-preta", não se consegue fazer com ele nada além do que ele foi programado para fazer, o que não promove o diferencial competitivo muitas vezes desejado.

O Yahoo e o Google utilizam uma parafernália de soluções tecnológicas, livres e proprietárias, pois só assim conseguem ao mesmo tempo otimizar recursos e buscar alguma vantagem competitiva além do que uma "caixa-preta" oferece.

Vale a pena ressaltar que *software* livre, como *software*

proprietário, não define um tipo específico de licenciamento. Existem, sim, várias licenças livres, mas que são ordenações jurídicas que devem ser respeitadas como em qualquer outra situação. Existem ainda *softwares* livres que, em certas condições, são passíveis de pagamento de licença de uso, normalmente a preços bem mais baixos que *softwares* proprietários equivalentes. Mas *software* livre também tem contrato, que ninguém lê e conhece, da mesma forma com que são tratados e pirateados os *softwares* proprietários. E é a isso que denomino a "pirataria do *software* livre".

## Aplicações Maduras de Softwares Livres

Existem alguns produtos livres que têm se destacado e substituído com vantagem alguns produtos proprietários, ora com mais recursos, ora com menos recursos, e sempre com menor valor de licenciamento e relicenciamento (*upgrades*), que é o que "pega" a longo prazo.

### 1) *software* livre rodando em plataformas proprietárias

Muito comum na Europa, esta alternativa parte do princípio de que:

- a plataforma Windows no *desktop* é inegavelmente mais fácil para instalar e operar, e conseqüentemente mais econômica;
- o usuário comum necessita de funcionalidades presentes em aplicativos que podem ser encontrados tanto no mundo livre quanto no mundo alternativo proprietário, com grande vanta-

gem econômica e de segurança.

Um situação típica seria um *desktop* com OpenOffice, Mozilla Thunderbird e Acrobat Reader. Essa configuração atenderia uma grande massa de usuários, a custo zero ou bem mais baixo do que a aquisição indiscriminada de produtos *mainstream*, principalmente em larga escala e a longo prazo, quando teriam sido contabilizados todos os *upgrades* de *software* e em cujo período os

usuários e administradores teriam tido tempo para entender a fundo os recursos desses produtos.

Esse tipo de utilização de *softwares* livres e alternativos impõe nenhuma dificuldade de instalação e portanto não requer capacitação especial nem por parte dos administradores nem por parte dos usuários.

### 2) *software* livre em servidores *web*

O servidor *web* mais usado no mundo é o Apache, que é



livre e gratuito. Não existe justificativa técnica para se utilizar outro produto na maioria das situações. O produto ainda é complementado por outras soluções livres como o Proxy Squid e o analisador de tráfego Webalizer.

### 3) **software livre em servidores de correio**

Existem aqui várias opções livres, campeãs de mercado, como o Postfix, o Qmail e o Exim, além de vários produtos acessórios tais como anti-vírus (Clamav) e anti-spam (Spamassassin). Muitas pessoas discutem a eficácia destes produtos e como eles sobrevivem. Este assunto é extenso mas, em poucas palavras, podemos dizer que o mercado oferece inúmeros *cases* de sucesso e, para quem está efetivamente atrás de economia, isto deve bastar.

### 4) **serviços de mensagens instantâneas**

Hoje existem formas de se conectar a qualquer serviço de mensagens proprietário (MSN, ICQ, etc.) através de produtos livres similares e que se conectam simultaneamente a todos esses protocolos. As interfaces são muito semelhantes e muitas vezes produtos alternativos oferecem recursos adicionais. Desse produtos, os de maior sucesso são o Gaim e o Kopete.

### 5) **software livre em VoIP**

Como o setor de telecomunicação opera com protocolos conhecidos, não é impossível desenvolver soluções abertas. As implementações do protocolo SIP e do aplicativo Asterisk permitem que um bom profissional de TI implante uma solução VoIP que pode vir a custar alguns milhões de reais.

### 6) **software livre em bancos de dados**

Os bancos de dados livres têm mostrado uma evolução muito grande. O PostgreSQL, o MySQL (agora com quatro produtos), o Firebird e o SQL Lite oferecem uma gama de opções muito interessante.

Já existem hoje mais de 3.500 clientes SAP rodando com bancos de dados livres (MaxDB). Existe uma fartura de produtos e recursos associados para várias situações (não todas, obviamente) e que, com algum conhecimento técnico, permitem substituir soluções caras e complexas onde estas forem subutilizadas.

Investimento em capacitação em TI e nos produtos específicos pode levar uma organização a economizar grandes somas com licenciamento e *upgrades*.

O real domínio das tecnologias e a real compreensão de como as coisas funcionam

levam ao desenvolvimento de soluções engenhosas e adaptadas a projetos específicos. Havendo resistência a este tipo de pensamento e com abundância de recursos, o melhor mesmo é se render aos produtos mais ortodoxos. Não é uma questão de arriscar e sim de capacitar e entender.

Para bancos de dados existem soluções de *cluster*, soluções Olap (proprietárias e alternativas), ferramentas para extração de dados e para desenho de diagrama entidade-relacionamento.

### 7) **desktops para telecentros**

Um bom exemplo disso são os telecentros da cidade de São Paulo, que operam com um servidor e estações (*desktops*) sem hd (*hard disk*). Nessa situação é muito interessante o uso de uma estação sem disco pela sua facilidade de configuração e administração. Foi justamente com essa possibilidade de operação que o sistema operacional de rede Novell Netware se firmou no mercado. Em larga escala, essa configuração de telecentros pode se tornar bem mais econômica em termos de *hardware* e de custo de administração, visto que é tudo centralizado nos servidores.

### 8) **e-gov**

Padrões interoperáveis são parte intrínseca de qualquer



produto livre. Documentos em formatos abertos (RTF e PDF) e programas escritos em linguagens multiplataforma (QT, Java, PHP, C) fazem parte do universo de produtos livres que podemos e devemos utilizar.

### 9) gerenciadores de impressão e de utilização de recursos de internet

Todo gestor de TI necessita controlar o volume de impressão por usuário (preço do cartucho!), uso de banda em navegação *web* e volume de *e-mails* recebidos e enviados. Estes são problemas triviais já resolvidos com produtos livres, tais como o Phpprintanalyze, Isoqlog e Sarg.

### 10) gestores de conteúdo

O *site* [www.cmsmatrix.org](http://www.cmsmatrix.org) cita mais de 100 produtos livres para desenvolver *sites*, cada um com características técnicas e comerciais (suporte, treinamento, etc.) diferentes.

### 11) servidores de arquivo

Aqui o Linux e o FreeBSD se destacam, dada a estabilidade e segurança comprovadas no desempenho dessas funções.

### 12) linguagens e ambientes de desenvolvimento para rodar em qualquer plataforma

As linguagens PHP e QT, entre outras, geram código que roda em qualquer sistema

operacional. Existem ainda ferramentas livres de apoio ao desenvolvimento (CVs, Bitkeeper) que controlam o desenvolvimento em equipe e *frameworks* de produtividade bastante avançados. Exceção se faz à linguagem Java, também multiplataforma, mas para a qual ainda é melhor usar Windows como ambiente de desenvolvimento (para rodar Java tanto faz) dada a estabilidade de funcionamento das ferramentas.

Poderíamos aqui estender esta lista a várias soluções, mas, com alguma possível omissão, são estas as situações em que mais se utiliza *software* livre.

## Racionalização de Recursos Aplicando-se o Conceito da "Máquina Atômica"

É comum encontrar um servidor com vários "serviços" (ou funções) instalados: servidor de arquivo, servidor de correio, servidor *web*, servidor de banco de dados, servidor de aplicativos, agentes *anti-spam*, etc. Cada um desses serviços consome recursos da máquina, o que faz com que o servidor seja mais parrudo, caro, muitas vezes com mais de um processador e vários hds. Além disso, é comum que esses serviços, operando juntos, acabem colidindo uns com os outros, diminuindo a confiabilidade do conjunto.

Os sistemas operacionais fechados não oferecem muitas opções de ajuste fino mas os abertos oferecem várias opções de *tuning* para aplicações diferentes.

No entanto, à medida que se ajusta o sistema para um serviço, desajusta-se para outro e então o único recurso é colocar mais memória, mais processador, mais espaço em disco, o que se traduz

em máquinas cada vez mais caras e de difícil reposição em caso de pane.

Surge aí o conceito da "máquina atômica", que é composto pela utilização de computadores convencionais (máquinas *commodity*), que podem ser adquiridas em qualquer lugar, rapidamente, a custo reduzido, e nelas se instala apenas um serviço. Para este único serviço instalado,

ajusta-se o sistema operacional especificamente para aquela função. Faz-se ainda um *backup* tipo imagem daquela máquina. Na eventualidade de algum problema, é feita a compra de outra máquina, coloca-se o CD com a imagem no *drive* e em poucos minutos a máquina está de volta no ar.

O mesmo conceito poderia ser aplicado a sistemas operacionais



proprietários mas, dado o custo de licenciamento e relicenciamento perpétuos, isto se traduziria em altos custos de TI. Fora isso, não se justifica um sistema operacional pago para esse

tipo de função. Alguns fabricantes de produtos proprietários como a Oracle até preferem o Linux.

É com esta arquitetura de "máquina atômica" que hoje

operam os grandes provedores, *data centers* e muitas empresas. É o conceito de *downsizing* levado ao extremo, onde tudo é barato, simples, focado e elementar, como um átomo.

## Situações e Cenários Favoráveis ao Uso de Software Livre

### 1) resolver problemas pontuais

É difícil generalizar mas *software* livre se encaixa muito bem quando existe um problema bem definido a ser resolvido. Os produtos proprietários na mesma categoria tendem a oferecer mais recursos, nem sempre utilizados. Se existe foco, definição, então é bem provável que o universo dos *softwares* livres

lhe atenda e muito bem. Se o problema é genérico, como um aplicativo que deve ter muitas funcionalidades, normalmente o mundo proprietário oferece melhores opções.

### 2) mão-de-obra capacitada

Se os profissionais entendem mesmo das arquiteturas e dos meandros da rede, se possuem a habilidade técnica de

"entortar" produtos, se têm prazer em fazer mais por menos, se têm prazer em pesquisar, se entendem bem inglês, então o caminho dos *softwares* livres será tranquilo.

Se, ao contrário, a sua turma é do "avançar, avançar, terminar", da interface gráfica para qualquer finalidade, então seguramente *software* livre não é sua praia.

---

## Conclusão

Tanto os *softwares* livres quanto outros produtos alternativos oferecem um grande potencial para economias e domínio tecnológico. Na ausência dessas necessidades e não havendo necessidade ou pressão para economias, os produtos proprietários cumprem melhor seu papel.

O custo relativo de *software* no Brasil é bem superior ao custo nos seus países de origem, devido a câmbio e impostos. Além disso, o custo de mão-de-obra aqui é

inferior. Esses dois fatores fazem com que tenhamos que repensar as análises de TCO feitas no exterior.

Nem sempre produtos livres ou alternativos são a melhor solução. Cabe aqui a análise de cada caso, estudo criterioso e, em última análise, uso do poder de decisão do gestor de TI e não do usuário. O usuário deve ser ouvido, mas não é ele quem paga a conta e não é ele quem pondera outros fatores que compõem as variáveis da equação da decisão.



Divulgação

## A Economia Java

**Michael Yuan** (michael.yuan@sun.com)

Engenheiro de Computação pela Universidade Estadual de Campinas. Arquiteto sênior de soluções corporativas baseadas em tecnologias Java e Web Services na Sun Microsystems do Brasil e evangelista de tecnologias Java desde 1999.

### RESUMO

Em maio de 1995, John Gage, diretor do departamento científico da Sun Microsystems e Marc Andreessen, co-fundador e vice-presidente executivo da Netscape, subiram juntos ao palco da conferência SunWorld, anunciaram a disponibilização da tecnologia Java e que ela estava sendo incorporada ao Netscape Navigator, na época o principal navegador da internet. Esse anúncio foi o lançamento oficial da tecnologia que, depois de dez anos, se tornaria o fundamento de um mercado, avaliado em 120 bilhões de dólares anuais, em pleno crescimento.

Em 2005 comemoramos o décimo aniversário da tecnologia Java. O objetivo deste artigo é mostrar a trajetória da tecnologia Java ao longo destes dez anos, o que a direciona, as oportunidades de negócios que gera e como se tornou o alicerce para um mercado de tão grandes proporções.

A adoção da tecnologia Java ao longo de seus dez anos de existência tem sido fenomenal. Uma análise do *status quo* revela números impressionantes. É estimado que Java gera um volume de negócios de US\$ 120 bilhões anuais na indústria tecnológica. O mercado de jogos Java para telefones celulares é estimado em US\$ 3 bilhões e sete de cada dez

aplicações sem fio em construção são baseadas na tecnologia Java. Do lado corporativo, o mercado de *software* para servir aplicações Java é avaliado em US\$ 2.2 bilhões com US\$ 110 bilhões em gastos de TI relacionados. Mundialmente, existem 4.5 milhões de desenvolvedores de *software* trabalhando com Java. Ao mesmo tempo, 100 operadoras de

telefonia móvel e 579 milhões de telefones celulares utilizam a plataforma e aproximadamente 825 milhões de cartões Java estão em uso globalmente. Estes números são uma amostra das proporções e do potencial deste mercado que denominamos "Economia Java".

Este fenômeno pode ser explicado levando-se em



consideração a combinação de quatro fatores complementares: compatibilidade, criatividade, negócios e comunidade. A plataforma Java é ubíqua e sua ubiqüidade vem de padronização que garante ampla compatibilidade. A compatibilidade atrai

desenvolvedores, que promovem uma saudável competição usando criatividade. A indústria utiliza e trabalha a criatividade dos desenvolvedores para criar um mercado consumidor e gerar negócios. Consumidores que apreciam a ubiqüidade,

compatibilidade e criatividade adotam, então, a tecnologia e se unem à Comunidade Java. Este é o ciclo que vem alimentando a evolução da tecnologia Java e desenvolvendo uma economia em torno da plataforma.

### Um Princípio Tecnológico Básico

A plataforma Java se baseia no poder da computação em rede e no princípio de que um mesmo *software* deve poder ser executado indistintamente em diferentes tipos de plataformas de *hardware*, sejam computadores, dispositivos pessoais ou outros aparelhos eletrônicos. Desde o seu lançamento comercial em 1995, a tecnologia Java cresceu grandemente em popularidade e uso, principalmente devido à sua real portabilidade, permitindo a execução de uma mesma aplicação Java em diferentes tipos de plataformas de *hardware*.

Qualquer aplicação Java pode ser facilmente transmitida através da internet ou de qualquer rede sem problemas relacionados a compatibilidade de plataforma de *hardware* ou sistema operacional. É possível, por exemplo, executar uma aplicação baseada em tecnologia Java em um PC com Windows, Linux, um computador Mac, uma estação Unix ou até mesmo em pequenos dispositivos como telefones celulares e

computadores de mão. O *slogan* "Write Once, Run Anywhere" (escreva uma vez e execute em qualquer lugar) é até hoje o grande apelo da tecnologia Java. Portabilidade é uma importante justificativa para adoção da tecnologia e, também, uma das principais razões pelas quais muitas grandes corporações a têm estabelecido como principal plataforma de desenvolvimento, motivadas pelo fato de que traz independência de fornecedores.

Uma outra importante característica da plataforma Java é o seu mecanismo de segurança. A plataforma foi desenhada para executar programas em rede, de forma segura, se integrando com segurança a sistemas existentes em uma rede. O mecanismo propicia controle elaborado de permissões para acesso a recursos de sistema e serviços, com fortes políticas de segurança, e mecanismos de criptografia que, de acordo com o histórico da tecnologia, têm oferecido uma barreira inviolável a qualquer tipo de

intrusão ou utilização indevida na plataforma.

Aplicações baseadas em tecnologia Java são programas escritos em linguagem de programação Java que podem ser executados em computadores pessoais ou em servidores, em grandes computadores *mainframe* ou outros dispositivos. Por exemplo, grandes companhias utilizam aplicações Java em servidores para monitorar transações e efetuar operações com dados existentes em outros sistemas computacionais. Outras companhias utilizam aplicações baseadas em tecnologia Java em seus *websites* internos para dinamizar a comunicação e o fluxo de informação entre departamentos, fornecedores e clientes. Há ainda outras que utilizam a tecnologia para criação de jogos *multiplayer*, que são executados em computadores pessoais conectados a servidores na internet.

Aplicações escritas na linguagem de programação Java podem ser executadas em



diferentes tipos de sistemas graças a um importante e poderoso componente da plataforma chamado máquina virtual Java, ou JVM. Ela pode ser entendida como um tradutor de instruções de plataforma Java para instruções próprias do dispositivo físico onde será feita a execução. A tecnologia da JVM evoluiu significativamente nestes dez anos. O que era apenas um interpretador de instruções Java foi se tornando um poderoso compilador em tempo de execução, com sofisticados

algoritmos de otimização e reutilização de memória, capazes de fazer com que aplicações sejam executadas com alta performance, muitas vezes superando a performance de aplicações desenvolvidas nativamente para um determinado dispositivo.

Este é o princípio da tecnologia: toda plataforma de *hardware* ou sistema operacional deve ser capaz de executar *software* desenvolvido com a tecnologia Java. E para isso, a tecnologia Java vem sendo incorporada em todo tipo

de plataforma durante estes últimos anos, desde celulares de última geração, aparelhos de TV digital, cartões inteligentes e muitos outros dispositivos pessoais, carros, eletrodomésticos, etc., além dos servidores corporativos nas empresas. Para as aplicações desenvolvidas com tecnologia Java não importa o tipo de computador, telefone, TV ou sistema operacional em que elas serão executadas: elas apenas funcionam em qualquer dispositivo habilitado com a plataforma Java.

## Uma História de Sucesso

A tecnologia Java foi criada como uma ferramenta de programação em um pequeno projeto fechado, iniciado por Patrick Naughton, Mike Sheridan, e James Gosling, funcionários da Sun Microsystems em 1991. A criação de uma linguagem de programação não era o objetivo principal do Projeto Green. O segredo Green Team era composto de 13 pessoas recrutadas pela Sun para antecipar e planejar a "próxima onda" da computação. O time chegou à conclusão de que uma das tendências mais significativas seria a convergência entre aparelhos controlados digitalmente e computadores. Então, para demonstrar o que visualizaram como o possível futuro dos aparelhos digitais, o Green Team se enclausurou durante 18 meses

para desenvolver o projeto. No verão de 1992, a equipe revelou um dispositivo de mão portátil, semelhante a um PDA, como conhecemos hoje, interativo e capaz de controlar aparelhos de entretenimento domésticos, através de uma tela animada, sensível ao toque, como interface de usuário.

O aparelho era capaz de controlar uma grande variedade de eletrodomésticos e plataformas de entretenimento – ao mesmo tempo em que mostrava animações – pois executava com uma linguagem completamente nova e independente de processadores. Esta linguagem foi criada especificamente para o dispositivo pelo membro do Green Team James Gosling que a denominou Oak (carvalho), por causa de uma árvore em frente à janela de seu

escritório. À medida que o projeto foi ganhando ímpeto e começou a envolver clientes em potencial na indústria de TV a cabo, o Green Team saiu de seu esconderijo e iniciou negociações para encontrar um mercado para a tecnologia criada no projeto. Aparentemente a indústria de entretenimento, especificamente, de aparelhos de TV por assinatura e vídeo sob demanda, parecia ser um mercado natural para essa tecnologia. O time, então, desenvolveu uma demonstração chamada MovieWood para demonstrar a tecnologia. Entretanto, essas indústrias estavam em sua infância e ainda estavam tentando se estabelecer como um modelo comercial viável.

"Naquela época, as coisas que fazíamos relacionadas a



tecnologia Java eram criadas ao redor de princípios de comunicação em rede, em um estilo bastante semelhante ao que é a internet", diz James Gosling. "Nós estávamos vendendo às empresas de TV a cabo a idéia de que era daquela forma que sua rede deveria ser. Ela deveria ser interativa e usuários deveriam ser capazes de ler e entrar com informação no sistema." Infelizmente, este discurso era muito avançado para a época. "Depois de nos darmos conta de que não havia negócios no mercado de TV digital por assinatura, precisávamos descobrir o que fazer com esta tecnologia ou, então, o que fazer com nossas vidas." Após um encontro de três dias, John Gage, James Gosling, Bill Joy, Patrick Naughton, Wayne Rosing e Eric Schmidt tiveram uma inspiração coletiva: por que não a internet? Afinal, a internet era exatamente o tipo de configuração em rede que o time havia vislumbrado para a indústria de TV por assinatura.

A internet estava se popularizando como uma maneira de movimentar conteúdo de mídia – texto, gráficos, vídeo – através de uma série de dispositivos heterogêneos utilizando HTML. A tecnologia Java havia sido desenhada para mover conteúdo de mídia entre redes de aparelhos heterogêneos, mas ela também oferecia a capacidade de mover "comportamento" na forma de

*applets* juntamente com o conteúdo. HTML não era capaz de fazê-lo, mas preparou o caminho para a tecnologia Java. Gosling explica: "Nós já estávamos desenvolvendo um tipo de 'roupa de baixo' para disponibilização de conteúdo ao mesmo tempo em que a *web* estava se desenvolvendo. Embora a *web* já estivesse presente por mais ou menos 20 anos, com FTP e Telnet, era difícil usá-la. Então veio o Mosaic em 1993, como uma interface de fácil uso para a *web* que revolucionou a percepção das pessoas. A internet estava se transformando, exatamente, naquela rede que nós estávamos tentando convencer as empresas de TV a cabo a construir. Tudo o que nós queríamos fazer, em sua generalidade, se encaixava perfeitamente na maneira como aplicações eram escritas, disponibilizadas e utilizadas na internet. Foi simplesmente um incrível acidente. Era evidentemente óbvio que a internet e Java eram uma combinação perfeita. Portanto, foi o que fizemos." O time desenvolveu um clone do Mosaic baseado em tecnologia Java que eles denominaram "WebRunner", conhecido mais tarde como o navegador HotJava. Era apenas uma demonstração, mas era algo realmente revolucionário: pela primeira vez, um navegador era capaz de exibir objetos animados em movimento e executar conteúdo dinâmico.

No começo de 1995, John Gage, diretor do departamento científico da Sun, foi convidado para falar em uma conferência de tecnologia, entretenimento e *design* que reunia profissionais de entretenimento e da internet do Vale do Silício e de Hollywood. No início da palestra, muitas pessoas pareciam distraídas. Afinal, não havia muita coisa interessante em uma nova linguagem que mostrava uma página de texto com ilustrações numa cópia do Mosaic. Então, James Gosling, passou o mouse sobre a imagem de uma molécula 3D, que rodou em todos os sentidos com o movimento do mouse. "Todo o público exclamou maravilhado, Ohhhhhh!", conta Gosling. "A visão de realidade daquelas pessoas mudou completamente porque a molécula se MOVEU."

Em seguida, Gosling e Gage levantaram a platéia com um algoritmo animado de ordenação de linhas que Gosling havia escrito. Aquela audiência jamais tinha visto nada além de imagens estáticas em um *browser*, mas ali as linhas se moviam como se puxadas por uma mão invisível. Instantaneamente, todos naquela sala repensavam o potencial da internet. A demonstração tinha impressionado um público muito influente, que aplaudia com entusiasmo – e graças a esse público, a notícia se espalharia muito rapidamente.



Em menos de um mês, a equipe disponibilizou o demo na internet para que um pequeno grupo de amigos e desenvolvedores o testasse. Pouco tempo depois, a equipe lançou uma versão totalmente pública do código-fonte Java na internet. Eles sabiam que liberar o código gratuitamente é uma das maneiras mais rápidas de gerar adoção em larga escala e também uma ótima maneira de obter a ajuda da comunidade de desenvolvedores para inspecionar código e encontrar *bugs*. O time criou então um endereço na internet, disponibilizando o código-fonte Java e, em poucos meses, foram mais de 10.000 *downloads* – uma surpresa para todos, inclusive para os criadores. Começaram a surgir requisições da imprensa, entrevistas, solicitações promocionais, convites para palestras e viagens por todo o mundo. Em breve a Sun

percebeu que a popularidade do time de tecnologia Java estava rápida e acidentalmente ultrapassando a sua própria e orquestrada popularidade, com virtualmente nenhum planejamento ou verba de *marketing*.

O momento foi propício para anunciar a tecnologia Java, pública e oficialmente. O anúncio foi agendado como parte de uma palestra de abertura na conferência SunWorld. Seria um anúncio breve, porém de grande impacto. A maior parte da equipe Java foi surpreendida com a presença de Marc Andeerssen, da Netscape, no anúncio. Eles não tomaram conhecimento de que a Sun e a Netscape haviam fechado um acordo na madrugada anterior ao anúncio: a tecnologia Java seria integrada ao *browser* Netscape Navigator. A perspectiva do Navigator ser capaz de abrir estes novos, misteriosos e simples

*applets* baseados em tecnologia Java em qualquer plataforma era uma notícia inacreditável para aquela influente audiência de tecnólogos e jornalistas – em menos de três minutos, todos aplaudiam fervorosamente, sabendo que testemunhavam o início de algo muito grande no futuro da internet.

Desde sua introdução em maio de 1995, a plataforma Java tem sido adotada pela indústria mais rapidamente do que qualquer outra tecnologia na história da computação. Todos os maiores fornecedores de plataformas computacionais assinaram contrato para integrar a tecnologia Java como parte essencial de seus produtos. Como disse Scott McNealy, CEO da Sun, para uma audiência numa conferência para a indústria: "Java se tornou a linguagem, a plataforma e a arquitetura para computação na rede."

## Trabalho Comunitário

Um dos pontos-chave da tecnologia Java e fator fundamental para o sucesso da plataforma é o aspecto da compatibilidade. Só pode haver ubiquidade de aplicações desenvolvidas com tecnologia Java se houver ubiquidade de plataformas para execução das aplicações. Em outras palavras, a plataforma Java precisa estar presente em todos os ambientes para que se possam executar aplicações

em todo lugar e, para tanto, é necessário compatibilidade. Ubiquidade é resultado de compatibilidade e compatibilidade é assegurada por padrões e especificações.

Um importante aspecto da plataforma Java é o fato de ela não ser propriedade de uma única empresa, mas de uma comunidade. O desenvolvimento e evolução da plataforma Java foi, durante todos estes anos, o

trabalho de uma comunidade em constante crescimento. Esta comunidade é responsável pela ubiquidade da plataforma Java, já que é ela quem decide qual o próximo dispositivo a ser habilitado com a plataforma, onde ela deve estar presente, que novos serviços ela deve ser capaz de oferecer. Em outras palavras, é ela quem padroniza e especifica a plataforma Java.



O trabalho da comunidade é coordenado por uma organização aberta chamada Java Community Process (JCP). Todos os padrões na tecnologia Java são especificados por grupos dentro da JCP denominados Expert Groups, abertos à participação de qualquer empresa, instituição ou indivíduo. Apenas no Brasil existem 23 empresas e instituições e 27 indivíduos membros da JCP, cooperando para padronizar tecnologias na plataforma Java, de um total de 900 membros em todo o mundo. Um dos princípios da organização é cumprir a promessa *"write once, run anywhere"* junto aos desenvolvedores. Um padrão só é útil se houver um bom método para validá-lo, por isso os processos da JCP são organizados de forma a produzir, juntamente com cada especificação, uma implementação de referência e um conjunto de testes de compatibilidade.

A tarefa da JCP não é uma tarefa simples. Há um grande número de empresas trabalhando em conjunto, cada uma com seus próprios interesses e objetivos de negócios. Os processos da JCP precisam garantir igualdade e confiança, balanceando velocidade, consenso e compatibilidade. Com essas premissas, a JCP se organiza em dois comitês executivos com 16 participantes

cada. Um deles é responsável por decidir a padronização da plataforma Java em pequenos dispositivos e dispositivos móveis (Java 2 Platform, Micro Edition - J2ME) e o outro por decidir a padronização das plataformas Java padrão (Java 2 Platform, Standard Edition - J2SE) e Java corporativo (Java 2 Platform, Enterprise Edition - J2EE). Cada membro dos comitês executivos permanecem na posição por três anos e são escolhidos em eleições que ocorrem anualmente, onde qualquer membro da JCP é elegível. Atualmente o comitê executivo de J2ME é composto por Ericsson, IBM, Intel, Matsushita, Motorola, Nokia, NTT Docomo, Orange France, Philips, RIM, Samsung, Siemens, Sony-Ericsson, Sun, Symbian e Vodafone; e o comitê executivo de J2SE/J2EE é composto por Apache Software Foundation, Apple, BEA, Borland, Fujitsu, Google, HP, IBM, IONA, Intel, JBoss, Doug Lea, Nortel, Oracle, SAP, Sun.

O processo de especificação de tecnologias Java se inicia com o envio de uma requisição por especificação Java (Java Specification Request - JSR) à JCP. Qualquer instituição, empresa ou indivíduo pode requisitar uma especificação descrevendo a proposta e a razão para desenvolvê-la. Dentro do processo de

especificação, a requisição é votada pelo comitê executivo correspondente e, se aprovada, inicia-se a formação do Expert Group. O próximo passo é elaborar um primeiro esboço da especificação, seguido de uma revisão. Nesse momento inicia-se o desenvolvimento de uma implementação de referência e de uma suíte de testes de compatibilidade. Ocorre então uma revisão pública da especificação e uma nova votação do comitê executivo. Se aprovada, a especificação é finalizada, juntamente com a implementação de referência e *kit* de compatibilidade, e ao final há uma última votação. Em caso de aprovação, a especificação é lançada oficialmente.

Este processo tem servido como alavancador e ao mesmo tempo guardião para a tecnologia Java. O maior risco para um padrão é o surgimento de diferentes implementações, como ocorreu, por exemplo, com o Unix e seus vários sabores e está acontecendo com o Linux e suas várias distribuições incompatíveis em sua maioria. A JCP tem tido sucesso em garantir que não existam diferentes implementações da plataforma Java, ao mesmo tempo em que tem concentrado os maiores especialistas na tecnologia Java em todo o mundo para evoluir e avançar a tecnologia.



## A Criatividade é o Limite

A cada ano novos usos interessantes da tecnologia Java têm mostrado que ela pode ser aplicada em praticamente qualquer necessidade e em praticamente qualquer situação. A criatividade dos desenvolvedores tem colocado Java em celulares, televisões, aparelhos de TV digital, equipamentos médicos e até mesmo em carros.

Recentemente, uma empresa americana especializada em monitores de saúde e bem-estar físico iniciou a comercialização de um interessante dispositivo móvel para monitoração de batimentos cardíacos. Há algum tempo, monitoração de pulso avançada só estava disponível em aplicações especializadas para militares em campo ou astronautas no espaço. A MedicTouch criou o sistema PulseMeter, que é composto de sensor de batimentos cardíacos, conectado a telefones celulares ou PDAs, em comunicação com servidores para armazenamento e análise de dados coletados. A tecnologia Java, como base para todo o sistema, é responsável por, desde a conectividade segura com dispositivos móveis, até a sofisticada interface gráfica nas telas dos telefones celulares e PDAs. A solução é bastante inovadora, pois transforma dispositivos utilizados prioritariamente para comunicação em ferramentas para monitoração remota, em tempo real, de saúde e bem-estar físico.

A tecnologia é utilizada também em situações extremamente críticas, como por exemplo, sistemas de segurança nacional. Pesquisadores de três laboratórios nacionais dos Estados Unidos, Argonne National Laboratory, Sandia National Laboratories e Lawrence Livermore National Laboratory, trabalharam em conjunto para desenvolver um sofisticado sistema de monitoração e alerta de emergência contra ataques químicos e biológicos em áreas de alto risco. O CB-EMIS (Chem-Bio Emergency Management Information System), já em operação nos sistemas de metrô de algumas cidades nos EUA, é um amplo sistema de vigilância que coleta dados de sensores, dados meteorológicos instantâneos, seqüências de vídeo e relatórios de trilhos em tempo real. Ele é capaz de até mesmo, quando necessário, executar modelos de dispersão de nuvens de agentes químicos para calcular trajetórias e início de atividade dos agentes. A decisão por utilizar a tecnologia Java para desenvolvimento do CB-EMIS foi, de acordo com pesquisadores dos laboratórios, devido à sua maturidade, escalabilidade e facilidade de manutenção.

Monitoração de poços de petróleo e gás é um interessante exemplo de solução baseada em tecnologia Java que leva em conta suas características de

comunicação em rede. A solução chamada Harmony coleta dados críticos de poços de petróleo e gás e os entrega em dispositivos móveis como telefones celulares. A informação, que anteriormente só estava disponível nos centros de controle das empresas de extração, agora pode ser acessada por gerentes de campo em locais remotos. Os dados podem ser acessados de diferentes maneiras. Os gerentes podem verificar o andamento da extração em um determinado poço ou o sistema Harmony pode enviar alertas para os gerentes de campo sobre problemas em algum local. As informações sobre os poços são visualizadas na forma de textos e gráficos na tela dos telefones celulares e os gerentes podem enviar instruções para operação nos poços através dos dispositivos móveis. A tecnologia Java é responsável pelo código compacto a ser executado nos telefones celulares, bem como em todos os outros dispositivos envolvidos no fluxo da informação – dispositivo de campo, estação de trabalho, navegador e servidor.

A tecnologia Java em breve estará também na casa de milhões de pessoas, permitindo que usuários de TV digital possam interagir com os programas de TV, participar de votações instantâneas, ver estatísticas e gráficos interativos durante jogos de futebol, comprar as roupas utilizadas por



apresentadores de TV durante um programa ou visualizar dados em tempo real sobre pesquisas de popularidade de candidatos durante uma eleição. O consórcio DVB (Digital Video Broadcasting), responsável por projetar e definir padrões globais para televisão digital e serviços de dados, é composto por mais de 260 operadoras de TV, fabricantes, operado-

ras de telecomunicações, engenheiros de *software* e organismos regulatórios em mais de 35 países. Em conjunto com a Sun Microsystems, o consórcio desenvolveu a plataforma MHP (Multimedia Home Platform), que é uma especificação de *software* baseada em tecnologia Java, a ser implementada em aparelhos de TV por assinatura, receptores de TV

digital e computadores pessoais multimídia para prover serviços de TV interativa digital. É interessante ver que a indústria inicialmente considerada pelos criadores da tecnologia Java como o mercado mais adequado para a tecnologia estaria, depois de dez anos, definindo-a como padrão para transmissão de TV interativa digital. Java está de volta às suas raízes.

---

## Conclusão

Em 1995 ninguém imaginou que a tecnologia Java, em dez anos, alimentaria computadores em aparelhos de TV, carros, aviões, foquetes e até mesmo no *rover* da Nasa, que coleta dados em Marte e os envia para a Terra. Hoje, a ubiqüidade da tecnologia Java é indiscutível. Ela traz interatividade à internet, gráficos em tempo real para aparelhos de televisão, imagens instantâneas em câmeras e jogos *multiplayer* para telefones celulares. Ela conecta as maiores corporações e as menores empresas aos seus funcionários, clientes e fornecedores, além de proteger um grande volume de transações eletrônicas no varejo, finanças, governo, pesquisa e medicina. Java está presente em todo lugar, efetuando transações financeiras, entretendo pessoas, coletando informações, monitorando saúde e fortalecendo as defesas de um

país. Anos atrás não se imaginava que a tecnologia Java nos ofereceria tantas possibilidades. A compatibilidade na plataforma, aliada à criatividade dos desenvolvedores, gerou uma economia e uma comunidade, sem precedentes, em torno da tecnologia Java. Os processos de evolução da tecnologia estão cada vez mais aperfeiçoados e acelerados; um número crescente de desenvolvedores empenha sua criatividade no desenvolvimento de soluções baseadas em Java; a cada momento mais empresas adotam a tecnologia como base para seus produtos e serviços e mais pessoas cooperam na evolução da tecnologia. Dez anos foram suficientes para que esta economia, fundamentada pela tecnologia Java, alcançasse tão grandes proporções. É motivador imaginar onde esta evolução pode nos levar em outros dez anos.



## Software Livre como uma Nova Forma de Geração de Conhecimento



Fotos: Foca Lisboa

### Wagner Meira Jr.

Professor do DCC / UFMG, doutor pela University of Rochester em Ciência da Computação. Publicou mais de cem artigos em fóruns internacionais e nacionais, além de dois livros. Pesquisador em produtividade do CNPq e consultor do projeto Internet 2 do Brasil, é um dos coordenadores do Projeto Tamanduá (plataforma de serviços de mineração de dados escalável e eficiente). Atua em projetos de *software* livre, como Agência Livre e LibertasBR.



### Marcio Luiz Bunte de Carvalho

Professor do DCC / UFMG, doutorando em Sistemas pela UFRJ, mestre em Pesquisa Operacional pela University of California em Berkeley. Sua área de pesquisa inclui projeto, implementação e análise de algoritmos dirigidos a problemas complexos. Membro do Laboratório de Software Livre da UFMG e do Conselho Editorial da Revista Fonte. Participa de projetos ligados à disseminação da internet no Brasil e educação a distância. É diretor de Tecnologia da Informação da UFMG.



### Dorgival Olavo Guedes Neto

Professor do DCC / UFMG, doutor em Ciência da Computação pela University of Arizona. Sua área de pesquisa se situa na combinação de elementos de sistemas operacionais, redes de computadores e sistemas distribuídos. Membro do Laboratório de Software Livre da UFMG, atua também no Centro de Acesso à Informação Digital (CAID) da UFMG, criado para ampliar a inclusão digital na Universidade.



### Renato Antonio Celso Ferreira

Professor do DCC / UFMG, doutor em Ciência da Computação pela University of Maryland, College Park. Pós-doutorado no Department of Biomedical Informatics da Ohio State University. Coordenador de Recursos Computacionais do DCC, atua em pesquisa em suporte a desenvolvimento e aplicações de alto-desempenho para plataformas paralelas e distribuídas. Pesquisador em produtividade do CNPq, atua como consultor Ad-hoc para o CNPq e CAPES. Membro do Laboratório de Software Livre do DCC.



## RESUMO

O *software* livre se apresenta como uma nova forma de produção de conhecimento, onde competição e cooperação caminham juntas, o que resulta em *softwares* mais robustos e que amadurecem mais rapidamente. Este artigo discute o processo de produção de *software* livre, apontando seus benefícios e riscos, além de descrever como um conjunto de *softwares* é integrado em uma distribuição, que é um instrumento padrão de entrega de *software* livre a usuários finais. Finalmente, descrevemos o projeto LibertasBR, onde se procura levar as estratégias de produção de *software* livre a outros ambientes de colaboração ainda não explorados entre os seus desenvolvedores e usuários, com vistas a facilitar ainda mais o uso e a disseminação do *software* livre.

O desenvolvimento da informática nos últimos 50 anos causou mudanças profundas nas formas de trabalho e de interação nas várias dimensões da sociedade. O computador é peça fundamental nesse desenvolvimento e consiste basicamente de uma infra-estrutura eletrônica e mecânica (*hardware*) que executa programas (*software*) a critério do usuário. Mais especificamente, o *hardware* provê um conjunto de operações que são compostas formando o *software* ou programas.

Embora a capacidade de processamento tenha aumentado de forma impressionante durante esse período, as premissas da arquitetura de *hardware* permanecem as mesmas que foram definidas por Von Neumann em meados do século passado. O *software*, por outro lado, tem se tornado cada vez mais complexo e de atuação mais ampla, tanto que a indústria de *software* já tem maior valor econômico do que a

indústria do *hardware*, onde o primeiro é executado. Um aspecto fundamental nesse progresso contínuo é o potencial de inovação inerente ao processo de construção de *software*. Construir um *software* é normalmente um processo de abstração altamente complexo, onde o desenvolvedor entende uma demanda real, constrói uma abstração para resolvê-la e materializa essa abstração na forma de um programa que é aplicado para resolver a demanda original. O aumento do poder computacional e de comunicação permitiu que abstrações cada vez mais complexas pudessem ser construídas, como interfaces gráficas e a própria internet.

Podemos caracterizar três dimensões de conhecimento envolvidas na construção e uso de um *software*: domínio de aplicação, programa e prática de uso. A primeira dimensão envolve o entendimento da demanda pelo *software* e do seu domínio de

aplicação, o que é o ponto de partida para a definição dos requisitos do *software*. A segunda dimensão compreende o *software* propriamente dito, envolvendo desde a definição do modelo de abstração que ele implementa até a seqüência de operações que ele executa quando invocado. A terceira dimensão é a habilidade de uso, ou seja, a capacidade de se utilizar o programa de forma apropriada, executando as tarefas para as quais o programa foi construído. Podemos entender bem essas dimensões se as analisarmos no contexto de um *software* muito popular: um editor de texto. Neste caso, a primeira dimensão consiste nas práticas de editoração eletrônica, como formatação, correção ortográfica e geração de índices, entre outras. A segunda dimensão é o conjunto de componentes de *software* que compõem o editor de texto e todo o seu processo de construção. A terceira dimensão compreende as



práticas de uso do editor, por exemplo, para escrever este artigo.

O ciclo de vida de um *software* pode ser dividido em quatro fases: especificação, desenvolvimento, implantação e manutenção. Durante a fase de especificação são determinados os requisitos do *software* a ser desenvolvido; mais especificamente, os recursos e funcionalidades a serem providos. Esses requisitos são o ponto de partida para o processo de desenvolvimento, onde os vários componentes do *software* são implementados, validados e integrados, até se obter uma versão que seja passível de utilização. Uma vez que esse ponto seja atingido, o *software* é liberado para testes e posteriormente uso (normalmente denominado versão "beta"), quando novos problemas são detectados e solucionados, culminando com o lançamento de uma versão estável do *software*. A partir desse momento se inicia o processo de manutenção, que tem por objetivo não apenas detectar erros, mas também detectar demandas para a evolução do *software*, quando começa um novo processo de desenvolvimento.

A produção de *software* é um processo intensivo do ponto de vista de mão-de-obra, a qual é normalmente especializada e de alto custo. O volume de *software* nos primeiros computadores era muito pequeno e o seu valor comercial reduzido quando comparado com o valor do *hardware*

de então. O resultado dessa relação de custos é que as pessoas distribuíam e principalmente trocavam *software* como mecanismo de aproveitamento mais eficiente do *hardware* de alto custo. À medida que o custo do *hardware* foi diminuindo e a complexidade do *software* aumentando, a relação foi se invertendo, a ponto do custo do *software* instalado em um computador ser maior do que o custo do *hardware*. Uma das consequências desse processo é o surgimento da indústria do *software*, que hoje já emprega mais do que a indústria do *hardware*, que se tornou altamente concentrada e automatizada.

O crescimento da indústria de *software* trouxe também a discussão a respeito dos modelos de negócio que a sustentariam. Um dos modelos de negócio mais populares nessa indústria é o licenciamento de *software* proprietário, onde empresas desenvolvem os programas e licenciam o seu uso, sem que os detalhes do *software* em si sejam publicados. Os programas nesse modelo se tornam segredo industrial e, portanto, definem a vantagem competitiva em relação a outros fornecedores, o que explica a postura de se restringir o acesso a esse conhecimento, seja por meios tecnológicos, seja por patentes. Nesse modelo de negócio, o custo de todas as fases do processo e mesmo de investimentos futuros é garantido pelas licenças.

Como o custo de reprodução de um *software* é muito baixo (o que não acontece na indústria de *hardware*), a lucratividade de uma empresa de *software* pode ser muito alta, principalmente se o *software* em questão é adquirido em larga escala. Um efeito colateral desse modelo de negócio é a concentração em poucos fornecedores, que ficam cada vez mais fortes em virtude da lucratividade advinda do volume. Mercados com essas características tendem a ser monopolistas e, infelizmente, já praticamente vivemos tal situação com relação a sistemas operacionais para computadores pessoais.

Um modelo diferente de produção e distribuição de *software* que, na verdade, foi a prática original na área de informática é o modelo de *software* livre. *Software* livre é um programa que os usuários podem executar, copiar, distribuir, estudar, modificar e aperfeiçoar, uma vez que o código-fonte é distribuído livremente junto com o programa resultante. Na sua definição mais abrangente, a única coisa que não se pode fazer com *software* livre é restringir esses níveis de liberdade. Esse conceito de liberdade é materializado na publicação irrestrita do *software* como mecanismo para a sua construção e aperfeiçoamento, que ocorrem de forma colaborativa, com centenas ou mesmo milhares de desenvolvedores utilizando a internet como meio de publicação, interação



e suporte ao desenvolvimento. O processo de produção de um *software* livre se inicia com um grupo de desenvolvedores especificando os requisitos iniciais do que eventualmente será um produto de *software*. Esse grupo pode ou não compartilhar as suas especificações publicamente, mas em geral define quais os requisitos a serem atendidos inicialmente. Esses requisitos são o ponto de partida para o processo de desenvolvimento, que é feito de forma colaborativa e utilizando maciçamente a internet como meio de comunicação. Uma outra característica é que o processo se baseia na publicação de resultados parciais, isto é, partes do código, que vão sendo finalizados e integrados de forma colaborativa. Esses fragmentos de código são testados assim que publicados e o processo de validação também acontece de forma colaborativa e redundante, com vários desenvolvedores e testadores trabalhando nos mesmos componentes de *software*. Esse sistema vai amadurecendo o código de forma independente e distribuída. Nas iniciativas mais bem-sucedidas, as atividades são moderadas por um responsável de reputação reconhecida na comunidade de desenvolvedores, o qual coordena as ações para garantir a integração dos esforços. O que tem sido surpreendente é que, apesar do seu arrojo, esse processo de desenvolvimento tem se mostrado um grande

sucesso. São milhares de projetos de *software* em todas as áreas de atuação da informática, que se apresentam hoje em variados graus de evolução e disseminação. Tanto do ponto de vista dos desenvolvedores, quanto do ponto de vista dos usuários, o processo de adesão a um projeto de *software* livre a ser desenvolvido ou utilizado leva em conta fatores como dinâmica de evolução e tradição e competência dos desenvolvedores, além da própria aplicabilidade do projeto em si. Ou seja, leva em consideração o dinamismo do desenvolvimento e a qualidade do *software*. Essa estrutura auto-organizada garante a sustentabilidade do *software* livre, seja em termos de novas versões, seja em termos de outros *softwares* que derivem de uma versão do *software*, o que é permitido e incentivado pelo modelo de licenciamento do *software* livre.

Mais do que uma ideologia ou um modelo de autoria, o desenvolvimento de *software* livre criou e amadureceu uma estratégia de geração de conhecimento sem precedentes, onde colaboração e competição se aliam para obter produtos de qualidade com rapidez. A colaboração acontece através da divisão maciça de tarefas num processo hierárquico, mas altamente descentralizado. Nesse processo, os desenvolvedores trabalham de forma independente, escolhendo e executando tarefas de desenvolvimento de

*software* de acordo com seus interesses e capacidades pessoais, que são submetidas à avaliação de desenvolvedores mais experientes, passando por um processo de homologação, que resulta na sua incorporação ao *software* que é publicado. A hierarquia de desenvolvedores é baseada em um relação de confiança, que tem como critério a competência técnica e experiência prévia de cada participante, de tal forma que o *software* é refinado sucessivamente ao longo da hierarquia, chegando ao topo com uma qualidade que tem superado frequentemente a obtida em projetos de *software* proprietário. A competição pode ocorrer dentro de um projeto, quando vários desenvolvedores trabalham nos mesmos módulos, ou entre projetos que tenham o mesmo propósito. Essa redundância de esforços pode não parecer uma opção eficiente, mas funciona bem como um mecanismo de controle da qualidade do processo de desenvolvimento. Uma vez que os desenvolvedores têm acesso a todo o código desenvolvido por seus pares, um processo contínuo de auditoria permite a descoberta precoce de deficiências e erros introduzidos pelos pares, permitindo uma evolução e maturação de projetos a uma taxa mais alta que inicialmente poder-se-ia esperar.

O processo de desenvolvimento de *software* livre se baseia ainda em dois componentes bastante peculiares: satisfação



pessoal e voluntariado. A força motriz dos desenvolvedores é a satisfação pessoal, ou seja, o reconhecimento por ter produzido parte de um *software* que é utilizado por milhares de pessoas. A satisfação pode ser vista como um misto de vaidade e de senso comunitário, ou seja, de ser reconhecido dentro de uma comunidade essencialmente meritocrática por sua própria competência e esforço. A participação nos projetos acontece de forma completamente voluntária, bastando ter um computador e uma conexão à internet para se tornar um desenvolvedor de *software* livre. É comum desenvolvedores participarem de vários projetos, tendo a liberdade de sair de um projeto quando considerarem pertinente. É interessante notar que essas características possibilitam um me-

canismo automático de seleção natural de projetos, fortalecendo aqueles projetos mais robustos tecnicamente, melhor organizados e com maior aplicabilidade. Como os desenvolvedores são voluntários, o aumento do número de desenvolvedores em um projeto é uma importante medida de sucesso. Um outro aspecto interessante é que os princípios do *software* livre permitem essa seleção natural de uma forma diferente. Por exemplo, uma ocorrência comum é um mesmo projeto ser o ponto de partida de vários outros, por motivos de nicho de aplicação ou mesmo afinidade técnica. Uma vez que o código é livre, esse tipo de evolução é natural e até desejável, promovendo uma renovação do processo, da equipe e das premissas do projeto.

A internet é reconhecida como um meio de disseminação de informação sem precedentes, que mudou os hábitos e a forma de comunicar e interagir. O desenvolvimento de *software* livre leva esse conceito mais adiante, permitindo a geração de conhecimento sem restrições e da forma mais democrática que já se conseguiu materializar. Mais ainda, acreditamos que os princípios do *software* livre possam se aplicar a outras áreas do conhecimento, uma vez que os benefícios são evidentes, pois compartilhar conhecimento livre significa receber muito mais do que se provê. A seguir vamos detalhar como o *software* livre chega ao usuário final e apresentar um projeto de pesquisa que investiga o próximo nível de interação em torno do desenvolvimento e adoção de *software* livre.

## Softwares, Pacotes e Distribuições

A produção de *software* livre é um processo auto-organizado, onde a dinâmica de cada grupo de desenvolvedores determina a taxa de evolução do respectivo *software* e os seus direcionamentos estratégicos. Uma consequência dessa auto-organização é a dificuldade de consolidação de uma solução, ou seja, um conjunto de *softwares* que atenda às necessidades de um dado usuário, que não necessariamente entende ou quer entender as peculiaridades da evolução de cada *software*. Essa consolidação é realizada

através das distribuições de *software*.

Uma distribuição é uma coleção de *softwares* que têm algum nível de maturidade e atendem a requisitos mínimos de interoperabilidade com os outros *softwares* da coleção. Podemos distinguir dois tipos de *software* em uma distribuição. O primeiro tipo são os *softwares* considerados básicos, como o sistema operacional, o gerenciador de interface gráfica e bibliotecas de funções para a construção de outros programas. O segundo tipo são os vários

programas e aplicativos que executam sobre ou utilizam os *softwares* do primeiro tipo. Neste caso, é fundamental que tudo se encaixe, isto é, seja compatível para fins de execução. Por exemplo, um aplicativo não pode demandar uma função que não é provida por uma biblioteca de funções. Em suma, uma distribuição agrega vários projetos de *software* livre em uma solução computacional que esteja ao alcance e atenda aos requisitos de usabilidade dos usuários. A construção de distribuições também é



realizada por grupos de desenvolvedores e há distribuições comerciais (isto é, pagas) e não comerciais. As distribuições comerciais mais conhecidas são a Suse, a Red Hat e a recém-criada Mandriva (que nasceu da junção da Mandrake com a Conectiva). Entre as distribuições não comerciais, Debian e Slackware são as mais tradicionais, mas há um grande número de distribuições em variados estágios de maturação. Em ambos os casos, um modelo de negócios comumente empregado envolve a prestação de serviços de configuração, suporte e manutenção, assim como treinamento.

As distribuições são organizadas como um conjunto de pacotes, onde cada pacote contém um ou mais arquivos, que podem

ser programas executáveis, arquivos de configuração ou mesmo de documentação. Além do conteúdo em si, esses pacotes também contêm informações de dependência (ou seja, de quais outros pacotes eles dependem ou quais outros pacotes devem estar instalados para que possam ser instalados com sucesso), o que também é a base dos mecanismos de atualização de *software*. Um aspecto importante a ser ressaltado é que a organização por pacotes permite que cada usuário personalize, de forma fácil, a sua instalação a partir de uma distribuição, escolhendo os pacotes que contêm os *softwares* de seu interesse.

O advento das distribuições impulsionou violentamente a adoção do *software* livre, em

particular para usuários leigos, uma vez que a distribuição toma para si a responsabilidade da instalação e configuração dos diversos componentes de *software*, os quais já estão pré-compilados. Nesse contexto, vale ressaltar que o trabalho de empresas ou grupos de desenvolvedores responsáveis por uma distribuição é constante e de complexidade crescente, considerando o aumento do número de *softwares* disponíveis e a sua complexidade. Essas equipes trabalham continuamente, seja atualizando pacotes, seja incluindo novos pacotes na distribuição, sem contar com as demandas de atualização da distribuição como um todo, em geral motivadas por uma nova versão do sistema operacional ou sistema de gerenciamento de janelas.

## *LibertasBR: Novas Dimensões de Colaboração*

Apesar dos avanços constantes e da maturação do processo de geração de distribuições, o processo de escolha, instalação e configuração de uma máquina baseada em *software* livre ainda pode apresentar algumas dificuldades para leigos. Alguns fatores contribuem para essa dificuldade. O primeiro é a seleção da distribuição a ser utilizada, uma vez que as diferenças entre as distribuições muitas vezes não são perceptíveis à primeira vista e o usuário acaba realizando um processo de tentativa e erro até encontrar a distribuição que seja mais adequada ao seu perfil e necessidades. O segundo fator é

que as principais distribuições são maximalistas no sentido de que provêm um grande número de *softwares*, muitos dos quais de pouco interesse para a grande maioria dos usuários. A escolha dos *softwares* a serem instalados se torna então outra barreira para os usuários, embora a flexibilidade oferecida pelas distribuições permita que eles sejam incluídos a qualquer momento após a instalação. O terceiro fator é que as demandas dos usuários podem diferir significativamente, o que torna inviável tentar definir uma distribuição "universal". Finalmente, as similaridades entre as preferências e necessidades

dos usuários não são exploradas, uma vez que cada um é responsável pelas suas escolhas, as quais são realizadas na maioria dos casos, como mencionado, através de um processo de tentativa e erro. É interessante notar que, embora haja uma grande efervescência em termos da produção de *softwares* isolados, o mesmo não ocorre em relação ao compartilhamento das configurações que os usuários utilizam nos vários nichos. Por exemplo, a configuração de *softwares* para uso em uma escola fundamental ou um telecentro não difere muito de um caso para outro e, mesmo assim, não há mecanismos



que permitam a troca dessas experiências.

O projeto LibertasBR tem como um dos seus objetivos diminuir essa lacuna e tornar o processo de personalização de distribuições mais fácil e intuitivo. Um dos pilares do projeto é um portal WWW onde o usuário pode facilmente gerar uma distribuição personalizada, assim como o respectivo conjunto de arquivos, o qual é obtido através de uma ou mais imagens de CD. Além de racionalizar o uso de banda de comunicação, o portal permite gerar distribuições em conformidade com as necessidades do usuário. Um aspecto interessante desse processo é que, à medida que distribuições são personalizadas pelos usuários, é possível determinar as necessidades típicas de cada perfil de aplicação, uma informação fornecida pelo usuário (ao indicar o objetivo da distribuição, por exemplo). Com base nas informações coletadas cria-se uma base de conhecimento sobre pacotes mais utilizados para cada perfil, permitindo apresentar sugestões para os novos usuários em

função de seus objetivos declarados, o que reduz o esforço para gerar uma distribuição que se aproxime mais das necessidades de um dado perfil. Para tal, o projeto contempla a implementação de um portal de conhecimento que contém informações sobre perfis associados a cada usuário e suas seleções de pacotes. O conhecimento constante do portal é aperfeiçoado a cada personalização de distribuição gerada, estabelecendo assim um processo de colaboração indireta entre usuários, que trocam informações sobre as suas opções de pacotes e formam comunidades virtuais determinadas pelas similaridades em termos dos recursos de *software* utilizados.

Além da capacidade de personalizar e compartilhar distribuições, o projeto LibertasBR também investiga mecanismos para incorporar tanto novas versões quanto novos *softwares* sem que isso comprometa a interoperabilidade dos pacotes que compõem uma distribuição. Nesse sentido, o projeto desenvolve metodologias de homologação de *software* para sua incorporação na base de pacotes a ser utilizada

na personalização de distribuições. Essas metodologias determinam as regras para empacotamento de *software* e afins, além da especificação de dependências impostas pelos mesmos. A idéia é que esse processo seja tão automatizado quanto possível, permitindo uma maior escalabilidade e oferta de pacotes, além de uma maior participação da comunidade de desenvolvedores como um todo. Novamente, a tônica é investigar novas formas de interação, extrapolando as idéias de *software* livre para além do simples compartilhamento de código, para o desenvolvimento de práticas colaborativas.

O projeto LibertasBR é hoje uma realidade e as suas soluções computacionais são a base de diversos telecentros no estado de Minas Gerais e mesmo fora dele, além de estar sendo amplamente utilizado na UFMG e outras instituições. Além da sua aplicabilidade imediata, vislumbramos que as técnicas e tecnologias desenvolvidas dentro do projeto possam servir de base para a construção de um conhecimento livre e sustentável.

---

## Conclusão

O *software* livre se apresenta como algo que vai muito além de um modelo de licenciamento ou uma estratégia de produção de *software*. Ele é a materialização de uma nova forma de produzir conhecimento, onde o compartilhamento aparece como postura

fundamental, motivado por fatores como satisfação pessoal, reconhecimento e o fato de que se recebe mais do que se provê. A obtenção de um *software* robusto, confiável e com grande capacidade de evolução é uma consequência do processo de produção de

conhecimento que permeia todas as dimensões da criação do *software* livre. Por isso, acreditamos que ele seja uma tendência inexorável, que está mudando os rumos não só da área de informática, mas do uso de tecnologia da informação por toda a sociedade.



Fotos: Divulgação



## Por Que Escolher?

### Rinaldo Zangirolami

Diretor Geral de Assuntos Jurídicos e Corporativos da Microsoft.

### Ygor Colalto Valerio

Membro do Departamento Jurídico, com foco em propriedade intelectual.

### RESUMO

Partindo do reconhecimento sobre o valor que a tecnologia da informação representa para empresas, governos e sociedade, o artigo enumera uma série de considerações sobre os fatores que devem ser considerados no momento da escolha entre um *software* de conformação livre ou proprietária, enfatizando a importância da adoção de critérios objetivos e neutros. Compara peculiaridades dos modelos de negócios de desenvolvimento e apresenta resultados de pesquisas realizadas por institutos independentes sobre pontos que devem ser considerados. Apresenta números relativos ao Custo Total de Propriedade (TCO) e analisa informações sobre o aspecto de segurança oferecida pelos dois modelos.

O valor que a tecnologia da informação representa para empresas, governos e para a sociedade como um todo dispensa longas explicações. O acesso à informação, por si só, tornou-se a matéria-prima de maior valor estratégico para os tomadores de decisões nos mais diversos níveis existentes, sendo que o meio digital é o veículo por excelência deste vasto fluxo de informações. Diante desse cenário, é fundamental que as decisões quanto a quais soluções de TI implementar

ou qual plataforma adotar sejam o resultado de uma seleção serena, precisa e segura.

No quase infinito horizonte de possibilidades que o crescimento da indústria mundial de *software* tem disponibilizado a governos e entidades privadas, como escolher qual caminho seguir? Decisões deste porte estratégico não podem se basear em mitos, subjetividades ou conceitos apriorísticos, mas em métricas e fatos passíveis de comprovação, típicos, por

sinal, da indústria da tecnologia.

Obviamente, a capacidade de municiar-se com dados para escolher esta ou aquela plataforma varia de empresa para empresa. No campo privado, o administrador decidirá conforme sua própria convicção, tendo plena liberdade para escolher, independentemente das informações que colher, do preço ou de qualquer outro fator. Entretanto, em um contexto governamental, de trato da coisa pública, em que o que



está em jogo é o melhor emprego dos recursos dos contribuintes, a escolha deste ou daquele produto deve ser fundamentada por critérios vários, como técnica e preço, que só podem ser sopesados após análise das propostas apresentadas, segundo procedimento previsto em lei. Qualquer preferência *a priori* ou sem comprovada razão não prevalecerá<sup>1</sup>.

A mensagem que acompanha esse discurso é a de que há muito mais a ser considerado na escolha

da plataforma do que o preço pago pela licença de uso de um programa. Quando o produto é *software*, a decisão entre livre<sup>2</sup> e proprietário<sup>3</sup> é muito mais difícil do que pode parecer a princípio, motivo pelo qual defendemos, mais do que tudo, a existência de neutralidade no momento de escolha.

Consumidores se beneficiam, hoje, de uma indústria da tecnologia da informação bastante competitiva, obrigada a focar-se mais e mais nos desejos e

necessidades de seus clientes. Mas, para eles, a comparação entre possibilidades não será a mais acertada se impregnada por discussões acerca do papel da propriedade intelectual, da política implementada pelo governo ou por quem detém os direitos sobre este ou aquele produto. O que se deseja é que a escolha seja pautada por estrita racionalidade, analisando-se vantagens e desvantagens objetivas dos produtos em questão. Travemos a batalha no campo dos fatos.

## 1. Modelos de Negócio e de Desenvolvimento

Não há uma maneira correta de se produzir ou de se comercializar um *software*. Modelos de desenvolvimento, implementação ou comercialização evoluem e se modificam, sendo que determinados fins justificam a adoção deste ou daquele modelo.

Empresas desenvolvedoras de *software* proprietário baseiam

seu modelo de negócios na exploração econômica exclusivista dos códigos que desenvolvem<sup>4</sup>, sendo que o instrumento utilizado para que se atinja esse aproveitamento econômico é o contrato de licenciamento. Por meio dele, a detentora dos direitos de propriedade intelectual que recaem sobre o *software* autoriza seu cliente a

utilizá-lo, de acordo com determinadas regras e restrições. Não há, via de regra, possibilidade de acesso ao código-fonte, nem autorização para alterá-lo.

O desenvolvimento do código, por sua vez, segue um processo previamente estruturado e amplamente controlado, acompanhado por extensas etapas de

1 Assim entendeu, por unanimidade, o Supremo Tribunal Federal, nos autos da Ação Direta de Inconstitucionalidade 3059, sustentando liminarmente a aplicabilidade da Lei 11.871/02 do Rio Grande do Sul, que determinava que os órgãos da administração direta, indireta, autárquica e fundacional utilizassem preferencialmente "programas abertos, livres de restrições proprietárias quanto a sua cessão, alteração e distribuição". Em voto lapidar, assevera o Senhor Ministro Carlos Ayres Britto: "Em suma, se os *softwares* abertos são, de fato, os mais interessantes para a Pública Administração, que isso fique demonstrado em termos de mais alta pontuação quanto aos critérios objetivamente figurantes de normas editais".

2 Apesar da diferença técnica existente entre os termos "*Software Livre*" e "*Software de Código Aberto*" (OSS - Open Source Software), optou-se pelo primeiro por sua mais ampla divulgação. Importante, apenas, não confundi-los com *freeware* (*software* gratuito).

3 Geralmente se utiliza o termo "proprietário" para definir *softwares* cujo código-fonte é fechado e protegido de maneira tradicional pelas leis de propriedade intelectual.

4 Não há dúvida de que não se trata apenas do código, mas do *software*, termo mais abrangente que inclui a existência, por exemplo, de elementos não literais como o *look and feel* (veja-se o caso Lotus vs. Paperback). Entretanto, as diferenças existentes entre "programas de computador" e "*software*" passam ao largo da finalidade deste artigo.



testes pré-lançamento em ambientes os mais variados, de maneira que o resultado final seja um produto o mais confiável possível.<sup>5</sup>

Por outro lado, o modelo de negócios implementado na exploração econômica do *software* livre e de código aberto é fortemente marcado pela necessidade de se associar a implementação da solução à mão-de-obra necessária ao emprego do sistema.

É comum que se confunda o Linux com suas distribuições. Quando se fala em Linux, está-se

referindo apenas ao *kernel* do sistema operacional, ou seja, ao conjunto de instruções essenciais ao funcionamento básico do *hardware*. Sobre este núcleo, diversas empresas desenvolvedoras de *software* de código aberto criam aplicativos e programas de sistemas, sendo que ao conjunto composto pelo código novo e o *kernel* do Linux é dado o nome de "distribuição".

Entre as distribuições mais populares hoje, encontramos Red Hat, Debian, Mandrakesoft<sup>6</sup> e

SUSE<sup>7</sup>, criadas e mantidas por empresas com modelos comerciais bastante lucrativos.

O desenvolvimento do *software* livre é um processo bastante complexo e fruto da interação entre inúmeras comunidades existentes para tal fim. Geralmente descentralizado, absorve criações desenvolvidas em todo o mundo e por pessoas diversas, seguindo ora um modelo hierárquico (catedral), ora um modelo difuso (bazar). Não há um processo padronizado.

## 2. Dados

Considerando os modelos de negócio e de desenvolvimento apresentados, passemos a uma análise baseada em pesquisas realizadas por institutos inde-

pendentes, sobre os pontos principais a serem considerados na decisão estratégica sobre qual plataforma usar. Os dados aqui apresentados estão disponíveis

ao acesso de qualquer um, e podem ser encontrados segundo o índice de referências bibliográficas apresentadas neste artigo.

### 2.1. Custo Total de Propriedade (TCO)

Tendo em vista o fato de que sistemas de informática em geral são ambientes altamente dinâmicos, utilizados regularmente e a longo prazo, é uma enorme imprecisão restringir seus custos ao momento de aquisição da licença de uso e de implantação do

sistema. Outros importantes fatores devem ser considerados para que se obtenha uma avaliação verdadeira, tais como o custo de implantação<sup>8</sup>, mão-de-obra técnica, gerenciamento, suporte e custos relacionados à estrutura de *hardware*<sup>9</sup>.

A esta métrica ampla e inclusiva de todos estes aspectos dá-se o nome de Custo Total de Propriedade<sup>10</sup>, que agrega dados a respeito de todos os relevantes fatores de custo ao longo da vida útil de uma determinada estrutura de TI.

5 Os modelos de desenvolvimento entre empresas variam, como se poderia esperar. Na Microsoft, antes de ser lançado, o *software* passa por pelo menos cinco processos diferentes de aprimoração (alfa, beta1, beta2, *release to market* e *release to customer*).

6 Atualmente Mandriva.

7 Recentemente adquirida pela Novell.

8 *Deployment*.

9 Custos de *hardware* com a plataforma Unix, por exemplo, são notoriamente maiores do que com Linux ou Windows.

10 Total Cost of Ownership (TCO).



Diversas análises neste campo já foram conduzidas por respeitadas entidades de pesquisa internacionais<sup>11</sup>, de maneira que tomadores de decisão no campo da tecnologia da informação têm, hoje, um vasto material à sua inteira disposição para que possam decidir sobre bases sólidas a respeito de que rumo seguir.

Para nossa análise, centrada na discussão de custos entre *software* livre e *software* proprietário, será importante verificar qual a importância do preço de aquisição de licenças de *software* na

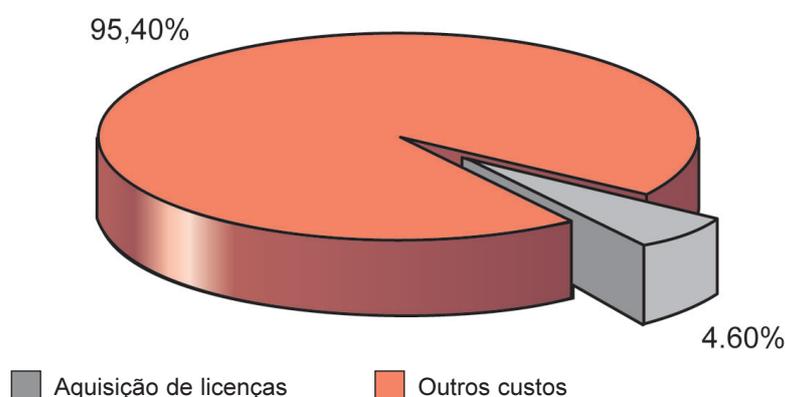
métrica do Custo Total de Propriedade, ou seja, considerando todos os gastos inerentes à manutenção de uma estrutura de TI, qual a representatividade do preço das licenças?

Este ponto é pacífico na literatura existente a respeito de TCO. Os mais variados institutos de pesquisa apontam que este custo específico tem um peso quase irrelevante. Com efeito, em estudo crítico sobre as diversas pesquisas existentes a respeito do tema, Alan MacCormack, membro docente da Harvard Business School, assevera que os "custos

de aquisição de *software* tendem a ser minimizados por outros custos, compondo, tipicamente, menos de 10% do Custo Total de Propriedade para um sistema. Isso sugere que o fato de um *software* ser livre, barato ou relativamente caro tem um baixo impacto relativo nos custos totais de investimentos em TI."<sup>12</sup>

Analisemos os dados coletados em pesquisa conduzida pelo IDC - International Data Corporation<sup>13</sup>, identificando qual a representatividade dos custos de aquisição de licenças num período de cinco anos:

#### Custo Total de Propriedade vs. Custo de Aquisição de Licenças



Conforme se verifica, segundo o instituto, a média de representatividade dos custos de aquisição da licença de *software* no Custo Total de Propriedade é de apenas e

tão somente 4,6%, tendo sido considerados operações de rede, sistema de arquivos, impressão, servidores de *web* e segurança.

O estudo detectou, ainda, qual

o maior componente na configuração do Custo Total de Propriedade de um sistema de TI: gastos com pessoal. O total atinge a casa dos 62%.

11 Entre as mais importantes: IDC, Gartner, D.H. Brown, Yankee Group, Forrester e Meta Group.

12 Evaluating TCO for Software Platforms: comparing apples, oranges and cucumbers. Harvard Business School, 2003. <http://www.aeibrookings.org/admin/authorpdfs/page.php?id=261>

13 Windows 2000 versus Linux in Enterprise Computing, IDC, 2002. <http://www.idc.com/getdoc.jsp?containerId=33131>



Diante dos números apresentados, constata-se que o fator em que o Windows é mais caro é precisamente o menos significativo. Por outro lado, o fator em que distribuições Linux são mais caras é justamente aquele que mais peso tem na composição do Custo Total de Propriedade de um sistema.

Confirmando a análise nos casos concretos do estudo, conduzido num período de cinco anos em 100 empresas, o IDC concluiu que as vantagens financeiras da escolha de uma plataforma Windows em vez de uma distribuição Linux variam entre 11% e 22%. Uma

economia extremamente significativa.

Outro estudo, realizado pela Forrester Research Inc.<sup>14</sup>, relata que o aumento de gastos com Linux varia de 5% a 10%. No mesmo sentido, há uma pesquisa realizada pela IWB em conjunto com a Universidade de São Paulo<sup>15</sup>.

## 2.2. Segurança: Linux é mais seguro do que Windows?

Foi a pergunta formulada e respondida pela Forrester Research Inc.<sup>16</sup>, com o intuito de analisar mais profundamente uma afirmação tida por muitos como verdadeira.

O estudo levou em conta uma comparação realizada entre o Windows e quatro das mais importantes distribuições de Linux existentes: Red Hat, Debian, Mandrakesoft e SUSE. A coleta

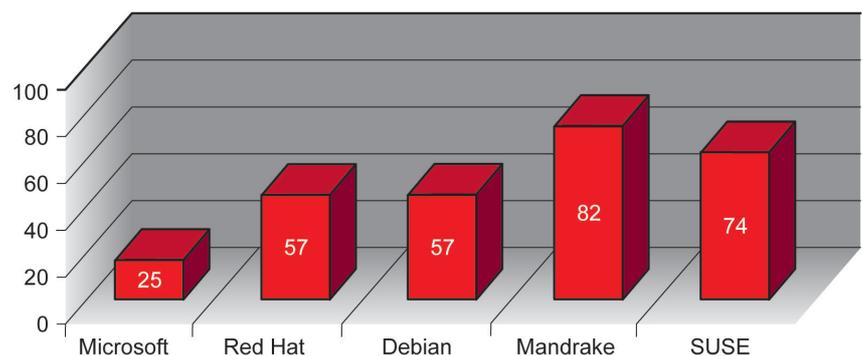
de dados, que teve por fonte informações "do mundo real", durou um ano, se estendendo de junho de 2002 a maio de 2003.

O método utilizado foi a contagem dos dias de risco dos cinco sistemas. A análise parte do pressuposto de que qualquer pedaço complexo de código é hipoteticamente vulnerável. Entretanto, esta vulnerabilidade não se

consubstancia em risco até que seja efetivamente descoberta, divulgada e explorada, sendo que os dias de risco se resumem ao tempo decorrido entre a divulgação da vulnerabilidade e a disponibilização do *patch* hábil à reparação da vulnerabilidade descoberta.

Os resultados divulgados pelo referido instituto de pesquisa foram os seguintes:

Dias de Risco



14 The Costs and Risks of Open Source, Forrester Research Inc. 2004.

<http://www.forrester.com/Research/Document/Excerpt/0,7211,34146,00.html>

15 Fogo Cruzado. IWB/USP, 2004. [http://www.itweb.com.br/solutions/gestao\\_empresarial/tco/artigo.asp?id=44063](http://www.itweb.com.br/solutions/gestao_empresarial/tco/artigo.asp?id=44063)

16 Is Linux More Secure Than Windows? Forrester Research Inc, 2004.

<http://www.forrester.com/Research/Document/Excerpt/0,7211,33941,00.html>



Conforme se pode verificar, a Microsoft obteve o menor número de dias de risco entre os cinco sistemas, colocando-se com uma vantagem de 32 dias em relação ao comparativo mais próximo, o Red Hat, que obteve uma média de 57 dias de risco no período analisado.

O fato de as vulnerabilidades serem corrigidas em um período de tempo significativamente menor pela Microsoft do que pelos produtores abertos

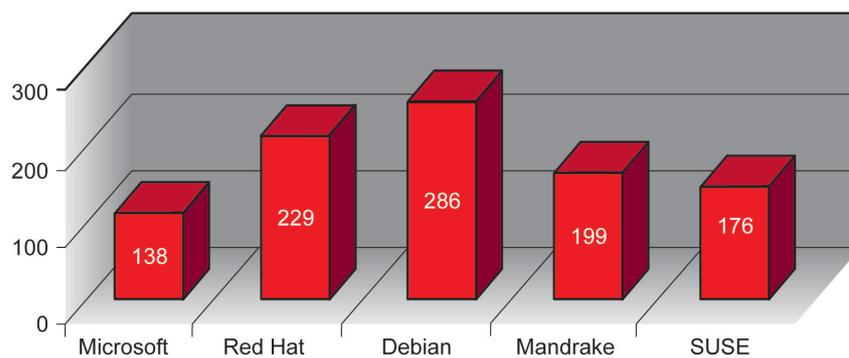
apresentados denota uma maior capacidade de resposta a questões de segurança. Uma questão já comentada anteriormente se apresenta como de importância-chave para o trabalho de proteção contra vulnerabilidades: o modelo de desenvolvimento de *software*.

A existência de um processo de desenvolvimento previamente estruturado e rigidamente controlado garante o sucesso de operações críticas, como a elaboração

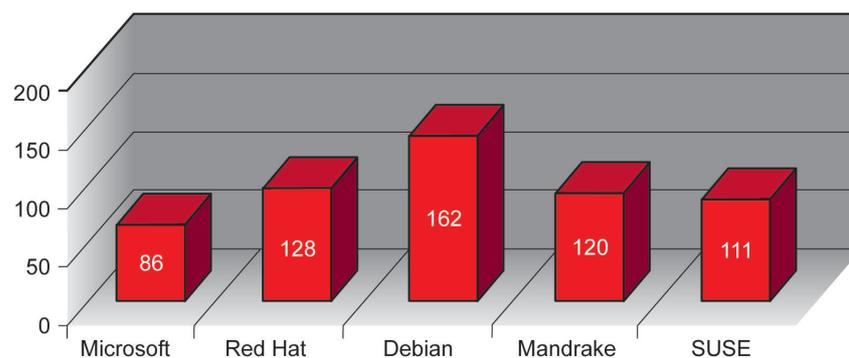
de um *patch* de segurança. A capacidade de direcionar a maior quantidade possível de esforços para um fim específico certamente será maior num ambiente corporativo do que em uma comunidade, o que se traduz em menor exposição a ataques.

Outro importante conjunto de dados foi analisado no estudo em referência: o número total de falhas e o número total de falhas críticas apresentadas. Os resultados:

Número Total de Falhas



Número Total de Falhas Críticas



A Microsoft obteve, portanto, a melhor colocação também no tocante ao número de falhas e ao

número de falhas críticas havidas, sendo também a única a corrigir 100% das falhas, seguida

por Red Hat, com 99,6%, MandrakeSoft com 99%, SUSE com 97,7% e Debian com 96,2%.



### 2.3. O risco da utilização de códigos de propriedade intelectual de terceiros

O tipo de proteção aplicada por cada nação varia, sendo que algumas optam por aplicar ao *software* o mesmo regime de guarda das criações artísticas e literárias – como o Brasil<sup>17</sup> – enquanto outras escolhem o regime paten-tário ou misto – como os EUA.

Em um ou em outro regime de proteção, o fato é um só: utilizar-se de código desenvolvido por outrem sem a devida autorização constitui infração à legislação de proteção à propriedade intelectual, dando ensejo à cobrança de indenização pelo detentor ori-ginário dos referidos direitos.

Ciente deste fato e desejosa de aumentar o valor agregado de

seus produtos, a Microsoft dá integral cobertura contra eventuais ações indenizatórias com base em violação de direitos de proprie-dade intelectual que sejam movi-das contra seus clientes. Respon-sabiliza-se completamente, por-tanto, pelo seu próprio produto.

A realidade para as platafor-mas livres não é a mesma, entre-tanto. Mesmo grandes empresas, como a IBM, não oferecem esse tipo de apoio aos consumidores de suas soluções baseadas em *software* livre.

O modelo de desenvolvimen-to influencia muito essa decisão. Por depender de criações elabo-radas no seio das comunidades

desenvolvedoras, não há uma maneira segura de se garantir que o código incorporado ao produto desenvolvido não foi criado por terceiro, o que torna claro o fato de a exposição a riscos deste tipo ser maior na implantação de plataformas baseadas em *soft-ware* livre. Além disso, a natureza aberta destas plataformas concede ao usuário a possibilidade de alterar o código-fonte, o que por si só já é um incremento ao risco de infringir propriedade intelectual alheia. Já há até empresas seguradoras oferecendo planos para o caso de o segurado ser condenado em uma demanda judicial dessa espécie.

---

## Conclusão

Diante dos dados demonstra-dos, fica claro que a decisão quan-to a que plataforma escolher não se resume ao custo de aquisição das licenças. A escolha é bem mais complexa do que se pode imagi-nar a princípio. Tomar a decisão mais correta dependerá sempre de uma postura inicial neutra, que permita uma análise real de van-tagens e desvantagens.

O que a indústria de *software* proprietário procura, hoje, é poder competir num mesmo patamar inicial, sem que haja preferências pré-definidas. Para ganhar a disputa no final, investe pesadamente em pesquisa e desenvolvimento,

com o intuito de incessantemente gerar mais valor para clientes e parceiros.

Mercados competitivos são o que há de mais saudável para o consumidor. Obriga empresas a man-terem seu foco nas necessidades e vontades de seus clientes, investi-rem no constante desenvolvimento de seus produtos e regularem seus preços de maneira a não perderem presença no mercado. Qualquer iniciativa governamental que bus-que prejudicar essa livre concor-rência deverá ser rechaçada veementemente, sob pena de se consti-tuir uma reserva de mercado cega à realidade dos custos e benefícios.

Das informações fornecidas neste artigo, duas conclusões são inafastáveis: *software* livre definitivamente não é gratuito e *soft-ware* proprietário não é, certa-mente, caro como se pensa. Com isso em mente, voltamos ao ponto de partida: com que base poderíamos preferir um a outro sem analisar os fatos?

O que se deseja é competir, concorrer. E a indústria proprie-tária tem, de sobra, valor, quali-dade e benefícios para oferecer a seus clientes, sejam do âmbito público, sejam do âmbito priva-do. Estamos aqui para oferecer aquilo que temos de melhor.

---

<sup>17</sup> Leis 9.609/98 e 9.610/98.



Lyderwan Santos

## Licenciamento de Software Livre: questões técnicas e legais

**Luiz Cláudio S. Caldas**

Professor universitário, formado em Engenharia Eletrônica e de Telecomunicações pela PUC MG e Direito pela UFMG. Mestrando na Faculdade de Direito Milton Campos, em Propriedade Intelectual no âmbito da Tecnologia da Informação. Analista de sistemas da Prodemge há 22 anos. Na Universidade Corporativa da Prodemge coordena a Gerência de Desenvolvimento e Estratégia Educacional.

### RESUMO

O artigo apresenta questões técnicas e legais relacionadas ao licenciamento do *software* livre. O conhecimento acerca das características e direitos permitidos pelas licenças de uso de *software* é ponto básico para a compreensão de sua forma de utilização.

### 1. Introdução

A área de tecnologia da informação e comunicação (TIC) tem sido um importante agente das transformações econômicas e sociais no mundo atual. No princípio da década de 70, já se destacava como um segmento econômico singular, ainda que restrito a operações entre as corporações que possuíam capacidade financeira para a realização de investimentos de maior vulto. Em face da alta especialização técnica requerida, representava um nicho de conhecimento restrito a um número relativamente pequeno

de empresas e pesquisadores. Com o aparecimento dos primeiros computadores de uso pessoal, a indústria experimentou um crescimento exponencial. Novas tecnologias propiciaram uma constante diminuição de preços das máquinas e simplificação nos seus processos de uso, popularizando rapidamente o segmento. O uso de computadores tornou-se corriqueiro, alcançando de maneira definitiva o dia-a-dia das empresas e das pessoas.

Na década de 80 surgiu no meio acadêmico um movimento

relacionado a um modelo diferenciado de desenvolvimento de *software*, que reivindicava a liberdade para acesso ao programa-fonte, pretendendo garantir a criação de produtos melhores e mais confiáveis. Potencializado pelas facilidades propiciadas pela internet e pelo crescente aumento de conexão entre os ambientes computacionais, a troca de programas entre essas pessoas criou as condições para a expansão dessas idéias, dando origem ao que ficou conhecido como "movimento pelo *software* livre".



Com forte motivação ideológica, a liberdade para distribuir o *software* desenvolvido, receber colaborações de terceiros e implementar sugestões provenientes de fontes diversas, sem maiores preocupações com o retorno financeiro, criou uma situação de oposição ao modelo tradicional adotado pela indústria, baseado na proteção fornecida pelos mecanismos oriundos da propriedade intelectual.

O tema *software* livre foi abordado de forma pioneira, em 1984, por Richard Stallman, na época, um pesquisador do MIT, através do anúncio do projeto GNU<sup>1</sup>, cuja idéia era a de construir um sistema operacional livre, compatível com o Unix. Stallman fundou a Free Software Foundation (FSF)<sup>2</sup>, com o propósito de divulgar as idéias acerca do movimento e dos conceitos que preconizava. Para proteger os produtos criados sob o ideal do *software* livre, a FSF estabeleceu uma forma engenhosa de proteção, através de uma licença de

uso conhecida como licença GPL, ou licença GNU/GPL, que, baseada nos conceitos jurídicos da propriedade intelectual e do *copyright*, estabelece que produtos criados sob essa forma de licenciamento não podem sofrer nenhum tipo de restrição quanto ao uso, cópia, distribuição ou liberdade de conhecimento do código-fonte.

O conceito de liberdade de *software* ganhou um impulso maior quando um estudante finlandês, Linus Torvalds, disponibilizou para críticas e colaborações o *kernel* de um sistema operacional compatível com o Unix (este de caráter proprietário). O sistema operacional, que ficou conhecido como Linux, recebeu melhorias, agregou funções, tornou-se robusto e confiável o suficiente para hoje estar instalado na maioria dos computadores que são utilizados como servidores de rede ou de banco de dados<sup>3</sup>.

O *software* livre e, principalmente, a discussão sobre a

extensão dos direitos da propriedade intelectual, criaram um caldo de cultura próprio e originaram em 1998 um outro movimento chamado *open source*, com ideais semelhantes, procurando adotar uma linha mais pragmática do que aquela defendida pelo movimento do *software* livre. Este grupo, liderado por Bruce Perens, John "maddog" Hal e Eric Raymond, originou a Open Source Initiative (OSI), entidade que hoje desempenha um papel destacado no movimento<sup>4</sup>.

A despeito de algumas diferenças existentes entre esses movimentos<sup>5</sup>, como veremos a seguir, adotaremos a denominação *software* livre para referirmo-nos a ambos.

O estágio atual do *software* livre não nos permite ignorá-lo: é um fato – e uma opção tecnológica – cada vez mais presente no cotidiano das empresas. Algumas vezes cercado por um viés ideológico, ainda suscita posições ardorosas, com defesas ou repúdios no mesmo diapasão. O

1 Richard Stallman trabalhava no Massachusetts Institute of Technology. É o fundador da Free Software Foundation e principal mentor do projeto GNU ("GNU's Not Unix").

2 Informações acerca da Free Software Foundation estão disponíveis no endereço <http://www.fsf.org>. Sobre o projeto GNU, veja em <http://www.gnu.org>.

3 Para maiores detalhes, veja o site da Linux International <http://www.li.org/>.

4 Informações sobre o movimento pelo código aberto (movimento open source) estão disponíveis no endereço <http://www.opensource.org>.

5 Os sites das entidades apresentam uma quantidade extensa de informações sobre elas. Recomendamos acessá-los, para obter informações detalhadas.



importante é compreender que o *software* livre, antes de receber a qualificação livre, é tão somente um *software*! E sendo apenas um *software*, pode possuir qualidades e defeitos inerentes a qualquer obra produzida pelo intelecto humano, que é capaz de transformar em linguagem de máquina processos intelectuais destinados à realização de determinado objetivo dependente de computadores para sua consecução.

Então, qual é a questão maior envolvendo o *software* livre, que provoca questionamentos e posicionamentos tão firmes quanto antagônicos?

Um ponto a ser considerado é o da proteção dos direitos da propriedade industrial. Questões legais envolvendo o *software* livre e a profusão de licenças criam por vezes uma situação complexa e incompreensível para muitos dos usuários e gerentes

de TI. As discussões sobre o *software* livre são frequentes e as opiniões extraídas desses debates são diversas e, às vezes, conflitantes. Essas discussões costumam convergir para a questão da propriedade intelectual e as implicações de cunho econômico, social e legal que daí decorrem. São aspectos que necessitam ser conhecidos para permitir que as decisões sejam tomadas de forma consciente e segura.

### 1.1. Os diversos aspectos da produção do software

A visão de desenvolvimento de um bem, que agrega valor aos negócios mas que não se transmite de forma onerosa, perturba o conceito vigente da produção de bem de valor econômico, protegido por instituto da propriedade intelectual, consoante os interesses econômicos daí advindos. O modo de produção capitalista, de uma maneira geral, não absorve, com facilidade, o conceito de produção de um bem que agrega valor a um negócio ou produto, no qual as pessoas envolvidas na sua produção não exigem, necessariamente, nenhuma espécie de contrapartida para o trabalho intelectual que despenderam. Com o *software* livre, o que acontece

na prática é a produção solidária e coletiva de obras intelectuais, por meio de pessoas que não se conhecem. Aparentemente essas pessoas trabalham de forma independente, com pouco controle centralizado e sem obedecer a nenhum tipo de hierarquia funcional. O resultado final desse esforço pode apresentar resultados úteis, que não pertencem a ninguém em particular, que poderão ou não ser comercializados por qualquer, se assim o desejar, independentemente da aquiescência daqueles que trabalharam na sua criação.

Vários trabalhos foram publicados nos últimos tempos procurando avaliar as implicações

econômicas e legais do *software* livre e as motivações das pessoas envolvidas com o movimento. Um texto interessante que descreve e compara a forma de produção de *software* livre e do *software* proprietário, intitulado "A catedral e o bazar", de autoria de Eric Raymond, tornou-se um clássico<sup>6</sup>. O filósofo catalão Manuel Castells possui uma obra muito interessante, "A sociedade em rede", na qual analisa os vários aspectos da internet e de sua influência na sociedade atual<sup>7</sup>. Não se centra no *software* livre, mas estuda de maneira ampla as implicações econômicas e sociais advindas da internet e do meio na qual está centrada.

6 O texto pode ser encontrado, no original em inglês, e em outros idiomas, no endereço <http://www.catb.org/~esr/writings/>.

7 O livro, em três volumes, foi publicado pela editora Paz e Terra e prefaciado por Fernando Henrique Cardoso, que observa que "Castells nos adverte,..., de que é preciso levar a sério as mudanças introduzidas em nosso padrão de sociabilidade, em razão das transformações tecnológicas e econômicas que fazem com que a relação dos indivíduos e da própria sociedade com o processo de inovação técnica tenha sofrido alterações consideráveis".



A internet<sup>8</sup> funciona como grande potencializadora desse movimento, na medida em que permite a troca de informações na grande rede, na maioria dos casos livre de restrições, controle ou limitações, seja de empresas ou de governos; está livre de censura de qualquer espécie, exceto os

limites impostos pelo próprio indivíduo. Na prática, estejamos em nossos escritórios, residências, centros de pesquisa ou *cyber-cafés*, podemos enviar e receber quaisquer tipos de informação, independentemente da existência de barreiras geográficas, idiomáticas ou econômicas.

Sob o aspecto legal, o *software*, livre ou proprietário, ainda é um tema não equacionado em sua completude, trazendo uma série de indagações a serem respondidas nas esferas relativas ao contrato, relações de consumo, responsabilidade civil e adesão das licenças ao ordenamento jurídico nacional.

## 2. A propriedade intelectual: breves aspectos

### 2.1. Apanhado histórico

Como instrumento formal e sistematizado, a proteção a uma obra intelectual aparece pela primeira vez na Inglaterra, em 1709, com a promulgação pela Rainha Ana do "Copyright Act", que concedia ao editor uma proteção com duração de 21 anos para as cópias impressas de determinada obra devidamente registrada. A lei não protegia a obra em si, mas as cópias que dela fossem feitas. O editor que registrasse uma obra recebia o privilégio para impressão e venda de sua cópia por determinado período de tempo.

Posteriormente, a revolução francesa amplia a concepção acerca do tema, contemplando outros

enfoques, além daqueles em voga. Surge a concepção do direito autoral<sup>9</sup>, estabelecendo privilégios ao autor. Mantém o aspecto patrimonial e introduz a esfera pessoal por meio dos "direitos morais" do autor.

O aspecto patrimonial relaciona-se à exploração econômica da obra, podendo ser transferido a terceiros. Os direitos morais, ligados ao autor, permitem-lhe certas prerrogativas, como a de reivindicar, a qualquer tempo, a paternidade da obra; de ter seu nome ou pseudônimo identificado nesta; de conservá-la inédita; de assegurar-lhe a integridade, opondo-se a quaisquer modificações ou à prática de atos que possam prejudicá-lo ou atingi-lo

em sua reputação e honra; de modificá-la, antes ou depois de circulação, ou de suspender qualquer forma de utilização já autorizada. É um direito intransferível e irrenunciável.

Foi um avanço significativo, porque o autor passa a ter seus direitos reconhecidos e adquire a capacidade para usufruir determinadas prerrogativas de caráter exclusivo.

No final do século XIX, com o crescente interesse econômico pelo tema, a doutrina começa a estruturar-se para tratar a propriedade intelectual sob um espectro mais amplo, objetivando compatibilizar legislações e ajustar o tratamento comercial. Começam

8 Ainda no prefácio de Fernando Henrique Cardoso, é "a sociedade em rede, globalizada e centrada no uso e aplicação da informação e na qual a divisão do trabalho se efetua, não tanto segundo jurisdições territoriais (embora isso continue também a ocorrer) mas sobretudo segundo um padrão complexo de redes interligadas".

9 Alguns autores posicionam-se entre as terminologias direito de autor e direito autoral. Para o propósito deste trabalho as denominações serão adotadas indistintamente.



a surgir tratados internacionais diversos, que se multiplicam continuamente à medida que novos campos do conhecimento adquirem valor econômico significativo.

Se a criação literária e sua reprodução é um fato antigo, a indústria fonográfica, de cinema e vídeo, a utilização de computadores, a biotecnologia e a genéti-

ca, por exemplo, são fenômenos recentes, que necessitam de novos entendimentos à medida que evoluem e assumem importância científica e econômica.

## 2.2. A propriedade intelectual sob a ótica legal

Cada país possui um ordenamento jurídico que lhe é peculiar e não é incomum existirem enfoques diferenciados para questões doutrinárias aparentemente semelhantes. Um determinado comportamento, uma determinada ação pode ser tida como legítima sob determinado ordenamento jurídico, mas considerada como delituosa em outro ordenamento. Adicionalmente, uma mesma questão, delituosa ou não, pode estar sujeita a um enquadramento jurídico em um certo lugar e ser tratada de forma diferente em outro. Assim é, por exemplo, a questão do *software*, que pode admitir em determinado ordenamento jurídico a proteção sob o regime do *copyright* e da patente e, em outros, como aqueles que sofrem influência do direito francês, receber proteção sob o regime de propriedade intelectual. São institutos que, em última instância, protegem as chamadas criações do espírito, mas possuem particularidades em sua forma de abordagem da questão.

A procura por um tratamento uniformizado do tema propriedade intelectual envolve a participação de diversos organismos internacionais que emanam recomendações para servirem como balizadoras no ajuste das legislações dos países signatários. O objetivo é uniformizar o tratamento dado às questões nos diversos sistemas legais. A importância do tema fez surgir no âmbito da ONU a Organização Mundial da Propriedade Intelectual (OMPI ou WIPO, em inglês), um fórum permanente para discussão de questões relacionadas à propriedade intelectual e articulação de políticas comuns.

Na explanação do prof. Denis Barbosa Borges, "a OMPI define a Propriedade intelectual como a soma dos direitos relativos às obras literárias, artísticas e científicas, às interpretações dos artistas intérpretes e às execuções dos artistas executantes, aos fonogramas e às emissões de radiodifusão, às invenções em todos os domínios da atividade humana,

às descobertas científicas, aos desenhos e modelos industriais, às marcas industriais, comerciais e de serviço, bem como às firmas comerciais e denominações comerciais, à proteção contra a concorrência desleal e todos os outros direitos inerentes à atividade intelectual nos domínios industrial, científico, literário e artístico. Tem-se, assim, correntemente, a noção de propriedade intelectual como a de um capítulo do Direito, compreendendo o campo da propriedade industrial, os direitos autorais e outros direitos sobre bens imateriais de vários gêneros"<sup>10</sup>.

A Comunidade Européia tem buscado compatibilizar suas normas internas por meio de um instrumento chamado Diretivas, um conjunto de recomendações que servem de base para orientação e ajuste nas legislações dos países-membros, uniformizando o entendimento e o tratamento jurídico em relação a determinado tema. Nas áreas relacionadas às tecnologias da informação e

10 Em relação à legislação brasileira, a lista completa dos temas protegidos por meio da propriedade industrial e de sua forma de proteção encontra-se no texto da Lei 9.279/96. Os elementos relacionados com o direito autoral estão explicitamente descritos nos termos da Lei 9.610/98.



comunicação já foram publicadas diversas Diretivas relacionadas com a proteção a bases de dados, *software*, telecomunicações, acesso à internet, políticas de privacidade de acesso e outros temas correlatos.

A propriedade industrial é o segmento que tutela as marcas, patentes, desenhos industriais e modelos de utilidade. O direito surge com o registro da obra em

órgão competente. O direito de autor relaciona-se à produção artística em geral, obras cinematográficas, músicas, filmes, ilustrações, artes plásticas, fotografia e programas de computador. O direito surge com a criação da obra (o registro é facultativo). A obra possui dupla proteção, através do direito patrimonial e do direito moral.

Em relação ao *software*, em

dezembro de 1996, a Conferência Diplomática de Genebra da Organização Mundial da Propriedade Intelectual aprovou o Tratado da OMPI sobre direito de autor, que versa sobre a proteção do programa de computador como obra literária<sup>11</sup>. É uma definição importante, pois afasta o programa de computador do outro ramo da propriedade intelectual, que é o da propriedade industrial<sup>12</sup>.

### 2.3. Aspectos da legislação brasileira

A legislação brasileira trata de forma explícita as questões relativas aos diferentes aspectos da propriedade intelectual. A propriedade industrial é disciplinada pela Lei 9.279/96. No art. 10º exclui, explicitamente, o *software* do escopo de sua proteção, deixando-a para legislação específica. O direito autoral é tratado genericamente pela Lei 9.610/98, que no seu art. 7º. informa que "os programas de computador são objeto de legislação específica..."

O programa de computador é objeto da Lei 9.609/98. A legislação brasileira inclui o programa de computador no regime de

proteção de direitos autorais, com alguma reserva em relação aos direitos morais, por força de convenções internacionais (Convenção de Berna e Acordo Trips). Há uma limitação aos direitos morais do autor do *software*, sendo reconhecidos apenas aqueles estritamente permitidos pela legislação.

Por meio desta sistemática entende-se que:

a) a legislação brasileira exclui o programa de computador do âmbito da propriedade industrial, por força explícita do disposto na Lei 9.279/96, que o coloca no âmbito do direito autoral;

- b) a lei de direitos autorais específica que o programa de computador é objeto de Lei específica;
- c) o programa de computador é regido explicitamente pelo disposto na Lei 9.609/98;
- d) a lei de direitos autorais possui aplicação subsidiária em relação ao *software*;
- e) outros instrumentos podem ser aplicáveis às questões legais decorrentes da comercialização e uso do *software*, como o Código de Defesa do Consumidor e o Código Civil.

11 José de Oliveira Ascensão, em sua obra Direito da Internet e da Sociedade da Informação, traz o texto do art. 4º. do Tratado de Genebra de 1966 em português: "Os programas de computador são protegidos como obras literárias no sentido do artigo 2 da Convenção de Berna. Essa proteção aplica-se aos programas de computador, qualquer que seja o seu modo ou forma de expressão".

12 Explica Newton Silveira que enquanto "as obras protegidas pelo direito do autor têm como único requisito a originalidade, as criações no campo da propriedade industrial,...., dependem do requisito de novidade, objetivamente considerado". (A propriedade intelectual e as novas leis autorais, Editora Saraiva).



### 3. Licenciamento de software

Um dos principais pontos da propriedade intelectual assenta-se na proteção dos direitos para a exploração econômica da obra. Seja qual for o mecanismo, por meio do direito autoral ou da propriedade industrial, a proteção é um imperativo econômico, social e legal. Disponibilizar o conhecimento em si sob qualquer forma, sem nenhuma restrição na ordem jurídica, permite a transferência deste conhecimento para o domínio comum da sociedade, possibilitando a qualquer um absorvê-lo, assimilá-lo e utilizá-lo da maneira que melhor lhe aprouver. É o que acontece, por exemplo quando as obras literárias e musicais caem em domínio público por exaustão do prazo de proteção: qualquer um pode comercializá-la, adaptá-la ou transformá-la da maneira que for conveniente.

A proteção dos direitos de propriedade intelectual sobre o

*software* se dá por meio de um mecanismo chamado licença de uso. A distribuição e comercialização do *software* proprietário faz-se em função do disposto em sua licença de uso. Ao adquirir um *software*, ao abrir a embalagem ou ao instalá-lo em um computador, o usuário, tacitamente, adere aos termos do contrato de licenciamento proposto pelo proprietário dos direitos sobre o *software*, sujeitando-se a sanções, caso descumpra quaisquer uma das cláusulas dessa licença. Normalmente, as licenças de uso do *software* proprietário não permitem cópia, redistribuição, nem permitem acesso ao código-fonte, que é tratado como segredo de negócio e é legalmente protegido.

O *software* livre também é protegido por licença de uso. E deve sê-lo, para fazer cumprir os ideais que propugna. Caso fosse

totalmente libertário e não fosse protegido por licença, se constituiria em uma obra sob domínio público, o que permitiria a qualquer um apropriar-se do código-fonte e incorporá-lo de forma proprietária a seus produtos. Para evitar esta situação, baseando-se nos conceitos da propriedade intelectual, os desenvolvedores gravam seus produtos com licenças específicas, onde as liberdades de uso estão garantidas.

Alguns produtos de *software* livre possuem mais de um tipo de licença. Uma destas segue os padrões das licenças livres e outra pode ser uma licença comercial. É o que se chama de *dual licensing*. O autor libera o código do produto de acordo com a situação; o usuário deverá adotar uma ou outra licença de uso. Exemplos clássicos dessa forma de licenciamento são o MySQL e o OpenOffice.

#### 3.1. A Free Software Foundation e a licença GNU/ GPL

Dentre as licenças de *software* livre, a licença GNU/GPL (GNU/General Public License ou, em português, Licença Pública Geral) é uma referência no setor e é a mais utilizada pelos desenvolvedores. Uma consulta ao *site* da Sourceforge (<http://www.sourceforge.org>), o maior repositório de produtos de *software* livre em desenvolvimento, mostrou que, do total de proje-

tos existentes, mais de 60% utilizam a licença GNU/GPL. Criada pela Free Software Foundation, esta licença baseou-se nos conceitos jurídicos do *copyright* para preservar o ideal do movimento.

A entidade, engenhosamente utilizou o enquadramento doutrinário da propriedade intelectual para introduzir o conceito de *copyleft*, em oposição aos conceitos do *copyright*. Enquanto o

detentor do *copyright* reserva para si todos os direitos relacionados com a exploração econômica da obra, o desenvolvedor do *software* livre, gravado com o conceito de *copyleft*, disponibiliza para toda a comunidade os direitos relativos àquela obra, com a restrição de que a ninguém será permitido apropriar-se daquele código, que, construído livre, deverá permanecer livre.



O raciocínio é simples: se o titular do direito autoral tem o poder para decidir a forma de utilização de sua obra, nada o impede de colocá-la à disposição de qualquer um, estabelecendo, inclusive, os limites e condições para uso e exploração de seu produto.

O conceito do *copyleft*, expandindo a utilização do produto original, permite a criação de obras derivadas a partir da obra principal, desde de que esta nova obra permaneça gravada com a licença original. Um produto criado a partir de um *software* que já possui a licença GNU/GPL também deverá possuir esta mesma licença. É o chamado *efeito viral* ou *caráter viral* da licença.

A Free Software Foundation assim conceitua o *software* livre:

*Software livre se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a quatro tipos de liberdade, para os usuários do software:*

- *a liberdade de executar o programa, para qualquer propósito (liberdade nº 0);*
- *a liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades (liberdade nº 1). Acesso ao código-*

*fonte é um pré-requisito para esta liberdade;*

- *a liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº 2);*
- *a liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 3). Acesso ao código-fonte é um pré-requisito para esta liberdade*<sup>13</sup>.

Os princípios básicos da licença GNU/GPL encontram-se descritos de forma clara e precisa no preâmbulo do seu texto<sup>14</sup>. O primeiro deles é o de que o *software* deve ser mantido livre, podendo ser modificado e distribuído pelo usuário sem necessidade de obtenção de licença do desenvolvedor. Um segundo, exige o distribuidor de qualquer responsabilidade sobre o produto distribuído. O terceiro é o de que a licença não permite a ninguém estabelecer qualquer tipo de proteção por meio de patentes, que possa transformar o código em proprietário.

Basicamente a licença apresenta como características:

- não é permitida a transformação de um *software* livre em proprietário;
- o conceito do *copyleft* (as liberdades) deve ser obedecido sempre;

- *software* livre não é necessariamente gratuito: nada impede que alguém estabeleça valor sobre um produto GNU/GPL, mas a obrigação de distribuir o código-fonte persiste;
- O produto é distribuído "*as is*", sem qualquer tipo de garantia;
- obras derivadas devem ser gravadas com a licença GNU/GPL.

A licença GNU/GPL é um padrão, que gerou várias licenças similares. A Free Software Foundation apresenta no endereço [http://www.fsf.org/licensing/licenses/index\\_html](http://www.fsf.org/licensing/licenses/index_html) uma relação das licenças que, sob a ótica da entidade, são compatíveis com a licença GNU/GPL. Apresenta também aquelas que, por alguma razão, são incompatíveis com a licença, mas mesmo assim possuem características de *software* livre e, ainda, uma relação de licenças não compatíveis com essa licença.

Objetivando atualizar a licença e compatibilizá-la com as demandas do mercado, a Free Software Foundation está trabalhando em uma atualização da licença GNU/GPL, chamada de versão 3. Até o presente momento, os termos dessa licença ainda não foram disponibilizados ao mercado.

<sup>13</sup> De acordo com o informado em <http://www.gnu.org/philosophy/free-sw.pt.html>

<sup>14</sup> Que podem ser conferidos, em português, no endereço [http://www.magnux.org/doc/GPL-pt\\_BR.txt](http://www.magnux.org/doc/GPL-pt_BR.txt).



### 3.2. A Open Source Initiative

A definição das características para que um produto possa ser considerado de código aberto (open source) estão elencadas na definição de propósitos apresentada pela Open Source Initiative (OSI). Basicamente, para que uma licença possa ser reconhecida como aderente ao padrão *open source* ela deve permitir a criação de trabalhos derivados, originários de um produto de código aberto, permitir uso comercial do produto e exigir que o código-fonte seja colocado disponível aos interessados.

Um *software* para ser reconhecido como aderente aos padrões preconizados pela Open Source Initiative deve atender aos critérios apresentados a seguir. Não existe uma "licença Open Source", nos moldes da licença GNU/GPL, mas um manifesto, uma declaração de princípios capaz de identificar as licenças aderentes ao padrão chamado de código aberto. A OSI se assemelha, em determinados propósitos, a uma certificadora, uma entidade que avalia se determinada licença adere aos padrões por ela defendidos. Esta declaração de

princípios se caracteriza pelos pontos elencados a seguir:

- liberdade de redistribuição: a licença não pode opor restrição a qualquer um que deseje redistribuir o produto, junto com outros *softwares*, independentemente da fonte. Não deve impor a exigência de pagamento, a qualquer título;
- código-fonte: a distribuição deve incluir o código-fonte e deve permitir a distribuição no formato aberto ou compilado. Caso o código-fonte não esteja sendo entregue, o responsável deve divulgar a forma de obtê-lo. O acesso ao código-fonte não pode ser obstaculizado;
- trabalhos derivados: a licença deve permitir modificações e a confecção de trabalhos derivados, garantindo a estes a distribuição sob as mesmas condições do trabalho original;
- a integridade do autor do código-fonte deve ser preservada;
- ausência de discriminação:

a licença não pode permitir a discriminação de qualquer espécie, em relação a pessoas, grupos, campos de pesquisa, ramos de conhecimento, etc;

- distribuição da licença: é uma cláusula que impede que o *software* seja fechado de alguma forma, perdendo seu caráter de liberdade;
- a licença não deve ser específica de um produto ou distribuição: se um programa ou parte dele é extraído de determinada distribuição, ele conserva a licença original;
- a licença distribuída com o *software* de código aberto deve respeitar as demais, não criando obrigações em relação a produtos;
- a licença deve ser tecnologicamente neutra.

O *site* da Open Source Initiative apresenta uma série de licenças que são consideradas aderentes aos propósitos definidos pela entidade. Na relação de licenças consideradas de código aberto está também listada a licença GNU/GPL.

### 3.3. Creative Commons

Nos últimos anos, inspirados nos conceitos de liberdade da criação intelectual, diversos trabalhos surgiram, buscando ampliar os conceitos do *software* livre para outros segmentos

representativos das criações intelectuais, como *software*, artes, música e literatura.

Em 2001, inspirando-se nos trabalhos desenvolvidos pela Free Software Foundation para o

mercado de *software*, um grupo de profissionais ligados a universidades norte-americanas (MIT, Harvard e Stanford), liderados por Lawrence Lessig, James Boyle, Michael Carroll e Davis



Guggenheim fundou uma entidade chamada Creative Commons (<http://www.creativecommons.org>).

A Creative Commons começou a desenvolver trabalhos em parceria com essas universidades, objetivando expandir o conceito do *software* livre a outras formas de bens culturais. O autor da obra, por meio de uma licença específica, poderia disponibilizá-la à comunidade, reservando para si os direitos que julgasse de seu

interesse. A entidade tem trabalhado no sentido de produzir licenças para os diversos segmentos: músicas, livros, *software* e artes em geral. A Creative Commons não possui uma licença exclusiva para o setor de *software*, mas um conjunto diferenciado de licenças que abrange outras modalidades de trabalho criativo como *websites*, música, filme, fotografia, literatura, etc.

A entidade não prega uma

ruptura completa com o sistema da proteção da propriedade intelectual, mas se propõe a permitir ao autor a liberdade para estabelecer o espectro de liberdade com o qual pretende gravar o seu trabalho. As licenças permitem que um autor de determinada obra, *software* inclusive, libere o uso de seu trabalho à comunidade, estabelecendo os limites dessa liberdade e reservando para si os direitos que julgar convenientes.

---

## Conclusão

O fenômeno do *software* livre é instigante e provoca discussões por força de paradigmas baseados em conceitos que são aceitos há alguns séculos. O *software* livre é um novo modelo de negócios que tem um nicho próprio e que vai se expandir até ocupar um limite razoável. Descobrir o meio termo, o ponto de equilíbrio entre essas forças pode ser ainda difícil, mas todos os atores terão que envidar esforços para encontrar este ponto. Não se pode esquecer que o *software* livre é um movimento maduro e grandes *players* do mercado mundial focam parte significativa de seus investimentos nesse mercado, através de unidades de negócio extremamente lucrativas.

Exemplos recentes mostram que mudanças tecnológicas forçaram a um realinhamento de negócios das empresas. Quando surgiram os primeiros aparelhos de vídeo, a indústria cinematográ-

fica temeu que esta tecnologia iria impactar de forma fundamental seus negócios. Com o passar do tempo, acabou por estabelecer uma forma de convivência, onde tecnologias de cópia e reprodução de imagens, como vídeos e DVDs, tornaram-se parceiros e agentes de expansão desta indústria.

A introdução da internet, com sua facilidade de comunicar e transferir informação, trouxe desafios para os negócios, economias, países e legislações. Disciplinar seu uso não é tarefa simples e ainda envolverá muita discussão.

Para uma avaliação madura da pertinência ou não da escolha de uma solução livre ou proprietária, as empresas devem considerar principalmente as questões relacionadas com a tecnologia, capacitação de pessoal e formas de licenciamento. Há maturidade do *software* livre em



diversos segmentos. Ambientes de servidores de redes e de bancos de dados possuem soluções estáveis e consolidadas, baseadas em *software* livre. Em outros ambientes, as soluções estão procurando um ponto de equilíbrio e há caminho a percorrer até que se possa dizer que as soluções em *software* livre estejam efetivamente maduras. Mas não se deve, porém, ignorar o potencial de amadurecimento das soluções dedicadas a esses ambientes.

Em uma lista não exaustiva, citamos alguns pontos que merecem uma avaliação criteriosa:

- a) o caráter viral da licença GNU/GPL, "infectando" outros *softwares* aos quais está associada;
- b) a existência de produtos com licença dual, que pode permitir interpretações errôneas acerca da utilização

- de uma ou outra licença em determinada aplicação;
- c) a questão da responsabilidade objetiva do desenvolvedor do *software*;
- d) o tratamento das relações de consumo e o conflito entre licenças e a legislação nacional, especificamente o Código de Defesa do Consumidor;
- e) o tratamento a ser dado a trabalhos derivados.

As discussões podem ser de vários tipos e dar-se de várias formas, mas o movimento e o conceito estão arraigados o suficiente para continuarem se desenvolvendo. Negá-los ou ignorá-los é perder uma oportunidade de participar de uma mudança que já está instalada.

"*Eppur si muove!*"<sup>15</sup>, teria dito Galileo Galilei. Esta frase tem aqui, também, um sentido bastante atual.

---

15 Esta frase teria sido dita por Galileo Galilei (1564-1642) após ter deixado a sala da Inquisição em abril de 1633, que culminou com sua renúncia às idéias sobre o heliocentrismo. Galileo foi considerado culpado de "grave suspeita de heresia" ao publicar obras nas quais afirmava ser o Sol o centro do Sistema Solar, em franca oposição à visão que colocava a Terra nesta posição. Em 1616 assinou um decreto do Tribunal da Inquisição, declarando ser meramente hipotético o sistema heliocêntrico. No *Diálogo dei massimi sistemi* (Diálogo sobre os grandes sistemas do universo) em 1632, voltou a defender o sistema heliocêntrico, tendo sido, novamente, submetido a julgamento e obrigado a negar a teoria copernicana. Ao sair do tribunal, teria dito a frase célebre: "*Eppur si muove!*", ou seja, "contudo, ela move-se", referindo-se à Terra.



## Evolução da Informática e Software livre

**Pedro Antonio Dourado de Rezende**

ATC PhD em Matemática Aplicada pela Universidade de Berkeley, professor de Ciência da Computação da Universidade de Brasília e representante da sociedade civil no Comitê Gestor da ICP-Brasil.

### RESUMO

O artigo trata das primeiras iniciativas de desenvolvimento de *software* livre e sua situação atual, enfatizando a questão do direito autoral e seus reflexos em questões de licenciamento, monopólio e exportação. Ressalta a necessidade de uma política para acompanhamento da evolução tecnológica, mais abrangente que a simples escolha do *software* livre como forma de redução de custos. Este artigo traça ainda um painel comparativo entre as diversas peculiaridades das regras negociais nos modelos livre e proprietário, e defende a autonomia tecnológica como argumento para adoção do FOSS como política pública.

A primeira proposta de um modelo para desenvolvimento de *software* livre, livre no sentido de tratar código-fonte como linguagem especializada, ao invés de segredo comercial ou industrial, surgiu com o projeto GNU, em 1984, e com a Free Software Foundation (FSF), em 1985. E as primeiras para licenciamento, em 1989, com o modelo de licença GPL da FSF, e com a licença BSD Networking Release 1 da Universidade de Berkeley.

Depois, em 1998, surge uma vertente mais pragmática e economicista, com a Open Source Initiative, modelando licenças com base na BSD. Juntas, as duas vertentes têm sido chamadas de FOSS (Free/Open Source Software). Alguns consideram o fenômeno do *software* livre um movimento tecnossocial; outros preferem vê-lo como uma etapa evolutiva da tecnologia da informação e outros, ainda, como aberração.

Considero-me um ativista do movimento desde 2000, por conta das primeiras jurisprudências abusivas do novo regime de direito autoral dos EUA, amparadas na Lei DMCA de 1998. Essas jurisprudências favoreceram, em detrimento de leis *antitrust* estabelecidas desde 1917, dispositivos esdrúxulos do DMCA para proteger a venda casada de DVDs e *softwares* proprietários, numa série de casos envolvendo autoria e distribuição do DeCSS, *software*



que permite o uso de DVDs legítimos em sistemas livres, esvaziando a doutrina do uso justo (*fair use*) sustentada na defesa.

O modelo FOSS vem ampliando seu horizonte de viabilidade, amadurecendo-se e consolidando-se. Principalmente no mercado de sistemas para servidores, onde em 2004 sua penetração cresceu, dependendo de como se meça, entre 34 e 50%. O modelo FOSS já havia provado sua viabilidade, fazendo acontecer a internet que conhecemos hoje, com servidores de correio (Sendmail, Postfix), de páginas *web* (Apache) e de nomes de domínio (Bind), predominantes em suas categorias. Mas é no *desktop*, na ponta do usuário final, último bastião do modelo proprietário, onde se protagoniza o conflito de interesses entre empreendimentos, ideologias e filosofias lastreados aos dois modelos.

Mais emblemático, para esse conflito, do que a jurisprudência envolvendo o DeCSS foi a chamada "guerra dos *browsers*" (navegadores). Nesta batalha jurídica que durou mais de cinco anos, a Microsoft foi acusada de prática monopolista predatória pelo Governo Federal e vários estaduais nos EUA e condenada em última instância pela corte suprema em outubro de 2001. Porém, após o juiz do caso ter sido

removido, por uma entrevista onde não se conteve e desabafou sobre o que considerou excessiva hipocrisia da ré, a pena estabelecida na sentença resultou praticamente numa espécie de prêmio.

Esta sentença gerou jurisprudência sobre direitos de licenciamento oneroso de padrões e formatos digitais, jurisprudência esta que os interesses monopolistas em TI querem agora estender a um regime radical e universal de propriedade intelectual, em acordos e fóruns internacionais, atropelando inclusive práticas e princípios da democracia, como recentemente na União Européia e na OMPI (Organização Mundial de Propriedade Intelectual).

Muitos interesses alinhados com os monopólios do modelo proprietário argumentam que uma política de informática favorável ao FOSS prejudica nosso esforço de exportação. Entretanto, "exportar *software*" ou "vender *software*" são expressões de veras vagas, já que podem significar quatro coisas distintas: vender licenças de uso, vender serviços, vender mão-de-obra especializada em programação ou vender o direito autoral de programas de computador. Os perfis tecnológico, educacional e empresarial da sociedade brasileira farão o Brasil em maior ou menor grau competitivo, em cada um desses quatro mercados

ou ramos do mercado global de *software*.

Ocorre que os dois modelos em questão dão ênfases distintas aos quatro ramos. Enquanto a venda de licenças é o carro-chefe no modelo proprietário, no modelo FOSS ela inexistente. Devido à natureza dos seus regimes de desenvolvimento e licenciamento, no modelo livre a licença de uso não é fonte de renda: é só, se assim lhe aprouver, geradora de mercado.

Em retrospectiva, podemos constatar que a natureza monopolista do modelo proprietário tem posto o Brasil, com seu perfil, em desvantagem no ramo das licenças de uso, a despeito de iniciativas como a do Softex, anteriores a 2002, como nos mostra nossa balança de pagamentos.

Entretanto, aqueles que lucram com tal situação engessam seu discurso anacrônico no plano político, como se o mercado de *software* fosse apenas o do ramo das licenças de uso. Com isso, enquanto buscam preservar suas posições, põem-se cada vez mais em defesa de interesses alheios ou mesmo antagônicos ao seu próprio desenvolvimento tecnológico e autonomia, frente às inexoráveis mudanças provocadas pela revolução digital.

Para corrigir essas distorções, de fato não basta o governo optar pelo uso de *software* livre onde



viável, gerando economia em licenciamento após investimentos na migração. É necessária uma política para acompanhar a evolução tecnológica. E se o governo se quiser soberano, essa deve pautar-se antes pelas vantagens que traga ao país e não ao capital abstrato em ambiente monopolizante por natureza.

Nesse tipo de análise, deve-se observar que a maior fatia do mercado para desenvolvimento de *software* não visa à distribuição e, portanto, não será nem livre nem proprietário, mas privado (*in house*). Ao passo que suas escolhas tecnológicas dependem hoje de um desses modelos, através do substrato de *software* básico onde irão operar seus produtos, substrato constituído de *softwares* distribuídos.

Há de se reconhecer, portanto, que não existe uma indústria de *software* livre per se, e sim atores que, na indústria de *software*, em maior ou menor grau, adotam ou tiram vantagens de um ou outro modelo em seus planos de negócio. Nesse contexto, a política de fomento que vejo como útil e vantajosa é antes pela mudança de enfoque no planejamento estratégico ou pela existência de um que melhor situe os agentes do mercado brasileiro quanto às perspectivas futuras, uma vez que a autonomia tecnológica só pode beneficiar nossa sociedade como um todo.

O Brasil já teve, antes do domínio do modelo proprietário e guardadas as devidas proporções, um quadro de mão-de-obra especializada em programação muito mais capacitado do que tem hoje. A Cobra, por exemplo, chegou a desenvolver nos anos 70 o quarto sistema operacional da família Unix do mundo, hoje abandonado. Se somos hoje mais consumidores do que produtores de *software*, em proporção maior do que já fomos, é lógico que seremos beneficiados por uma melhor relação custo/benefício, não o contrário. Os monopólios de TI querem pendurar a conta do combate à pirataria, com estridência crescente, no Estado, e isso precisa ser contabilizado na relação custo/benefício como custo social do modelo proprietário.

Muitos "especialistas" falam do mercado de *software* como se as regras negociais do modelo proprietário estivessem insculpidas na pedra de Moisés ou na das leis da economia, mas não é assim. Tais regras só se estabeleceram nos últimos 25 anos, depois que o negócio do *software* se descolou do negócio do *hardware*. Nos ciclos anteriores, modelos negociais predominantes nunca trataram código-fonte como segredo de negócio.

E na história da informática, que tem pouco mais de sessenta anos, esses ciclos têm mudado a

cada vinte anos, enquanto nada indica que o modelo proprietário seja o último. Pelo contrário, dada a internet. No contexto atual, com a internet, o desenvolvimento de *software* pôde evoluir bastante e poderá continuar evoluindo, em relação aos modelos anteriormente prevalentes.

O regime FOSS, sendo colaborativo, é mais eficiente se houver competência na gerência do projeto e motivação de participantes. Sem custos de policiamento das licenças de uso, que só aumenta com hiperconectividade, e sem estratégias artificiais de obsolescência programada, inúteis no contexto das liberdades que o caracterizam, a evolução de um *software* livre depende apenas do interesse de capazes em nele investir. E os recursos abundam quando o interesse é coletivo e o retorno indireto é socializado, em círculo virtuoso. A internet é o melhor caso de sucesso deste processo.

Entretanto, a razão mais premente para uma política pública em favor do FOSS é a autonomia tecnológica. Na história da informática, a separação do negócio do *software* em relação ao do *hardware* permitiu que empreendimentos de produção e distribuição de *softwares* passassem a se basear no controle do código-fonte, forma de expressão na qual são esses originalmente escritos.



A partir do código-fonte pode-se produzir, por tradução automática e replicação, exemplares em código executável em vários sistemas, com custo adicional quase nulo. As formas de controle, por sua vez, contrastam o regime hoje prevalente – o proprietário – com um emergente – o FOSS, do *software* livre ou de código aberto.

Para o regime proprietário, código-fonte é segredo de negócio, propriedade do empreendedor e em sua tutela. Já para o FOSS, código-fonte é conhecimento sedimentado em linguagem técnica, acessível a interessados e de livre circulação. Daí, métodos e processos de produção e de gestão e modelos contratuais ou negociais subordinados aos dois regimes se orientam por princípios distintos.

A licença de uso para executáveis, por exemplo, cujo comércio é fonte estruturante de renda no regime proprietário, torna-se cessão de direitos no FOSS. Cessão irrestrita de direitos não excludentes para usufruto de um bem não rival e, portanto, instrumento inviável como fonte de renda.

Assim, empreendimentos em *software* livre ou de código aber-

to precisam buscar fontes de retorno ao investimento distintas do licenciamento de uso, tais como suporte, customização e serviços, esses também explorados no regime proprietário. Num primeiro exame, parece impossível que o FOSS possa prosperar e competir, com um canal de renda a menos, numa economia de livre mercado.

Mas a revolução digital revela novas dimensões estruturantes da produtividade, que os regimes industriais e práticas mercantis tradicionais estão inadequadamente equipados para explorar e que os regimes emergentes buscam empiricamente desenvolver. Um dos principais contrastes entre o modelo FOSS e o proprietário está na inversão da dominância nas relações trabalhistas entre empreendedor e desenvolvedor de programas, podendo este ser inicialmente um simples e único programador.

Em artigo seminal, "Coase's Penguin", Yohai Benkler analisa, à luz de teorias econômicas e jurídicas em voga, como e por que isto é possível e viável, não só para a produção e dis-

tribuição de *softwares*, mas de qualquer bem para o qual o insumo e o produto são, em essência, informação e os meios se baseiam em conhecimento. Trata-se de um novo modo, pós-industrial de produção, que ele chama de "*common's based peer production*".

A grosso modo, a viabilidade do modelo FOSS está atrelada à abundância de código-fonte disponível, à eficácia crescente das técnicas e ferramentas de programação que exploram o reuso de código e à motivação como moeda de crédito para a produção. Crédito que tende a aumentar com abusos monopolistas no modelo proprietário.

De empresas com passado ou com vocação monopolista podemos esperar ou a reorientação de suas estratégias, como faz a IBM, ou o entrincheiramento na defesa do *status quo*, como faz a Microsoft. Quem tem mais a perder, a meu ver, com os ventos da mudança, são empresas monopolistas que apostam suas fichas na sustentação do *status quo*, resistindo aos efeitos da revolução tecnológica nas práticas sociais.



## *SGBD com Extensão Espacial e Sistemas de Geoinformação: Um Casamento Perfeito*

### **Cleber Hostalácio de Melo**

Engenheiro Eletricista (1984) e mestre em Ciência da Computação (1988) pela UFMG. Trabalha na Prodemge desde 1989 como analista de tecnologia, em prospecção e implementação de novas tecnologias de informação, com ênfase atualmente em banco de dados, recuperação de informações textuais e ferramentas e tecnologias associadas à linguagem de programação Java.

### **Márcio Azevedo de Menezes Guerra**

Engenheiro metalurgista (1979) e especialista em modelagem e simulação de processos siderúrgicos (1980) pela UFMG. Trabalha na Prodemge há 10 anos como analista de informática, com ênfase em geoprocessamento, prestando consultoria e desenvolvimento de aplicações SIG.



### **RESUMO**

O artigo inicialmente apresenta um breve relato da história e evolução da área de geoprocessamento, desde a década de 60 até os nossos dias, apresentando os principais conceitos e evoluções tecnológicas. Em seguida, ele descreve as arquiteturas básicas para se implementar a gerência de dados em sistemas de geoinformações e conclui, apresentando o resultado de um estudo comparativo realizado na Prodemge, envolvendo os seguintes SGBD objeto-relacionais e suas extensões espaciais: MySQL, Oracle e PostgreSQL.

**Palavras-chave:** Geoprocessamento, Sistemas de Informações Geográficas, SIG, Sistemas de Geoinformações, SGBD Objeto-Relacionais, Oracle, MySQL, PostgreSQL

### *Onde Tudo Começou?*

O ser humano sempre coletou informações sobre a geografia ao seu redor, procurando a existência de determinado recurso necessário a sua sustentação. Ao longo

do tempo essa coleta de informações evoluiu, incluindo novos elementos, tais como obstáculos naturais e recursos minerais.

A evolução natural do homem

para a organização em sociedade exigiu que essas informações fossem consolidadas e organizadas, para possibilitar a sua difusão e compartilhamento entre todos os



interessados. Começa a utilização de papel, escala e notações especiais para representar graficamente essas informações, na forma que ficou universalmente conhecida como mapa. Essa forma de representação possuía limitações inerentes como, por exemplo, combinar e analisar informações provenientes de diversos mapas distintos.

A partir de 1960, com a evolução na tecnologia de informação, tornou-se possível representar essas informações em ambiente computacional. Os dados passam a ser organizados de forma a facilitar a análise de seus inter-relacionamentos, com o objetivo de gerar novos dados e representá-los no espaço geográfico. Surge então o geoprocessamento. Utilizando técnicas matemáticas e computacionais, o geoprocessamento permite a correlação e a análise de dados geográficos e de dados alfanuméricos (seus atributos), facilitando a tomada de decisão em diversas áreas, tais como planejamento urbano e regional, gestão de recursos naturais, transporte, gestão nas áreas de Saúde e Educação.

No ambiente computacional, as ferramentas de *software* utilizadas para realizar o geoprocessamento são denominadas Sistemas de Informações Geográficas - SIG<sup>1</sup> (em inglês:

Geographic Information Systems – GIS). Essas ferramentas permitem análises complexas, combinando dados alfanuméricos com dados geográficos. Os dados geográficos são denominados temas ou camadas (em inglês: *themes* ou *layers*). Um determinado tema pode ser representado por uma ou mais camadas, dependendo da complexidade, da forma de representação dos dados ou de ambos. Exemplificando, a análise do deslocamento dos alunos para chegar a determinada escola pode envolver três camadas localizadas geograficamente ou seja, georreferenciadas:

1. a primeira camada contém as escolas existentes;
2. a segunda apresenta os endereços residenciais dos alunos e também será georreferenciada com base no cadastro de quadras e lotes do município;
3. a última camada representa as linhas de transporte urbano em operação.

Uma característica relevante dessa abordagem é que os dados em cada camada podem ser originados de diversas fontes distintas. No exemplo anterior, a primeira e segunda camadas seriam geradas pela Secretaria de Educação, a partir do Cadastro Técnico Municipal da Secretaria da Fazenda, enquanto a terceira

camada poderia ter sido originada da(s) empresa(s) de transporte municipal.

A partir de 1980, com a redução de custo e evolução da capacidade computacional do *hardware*, iniciou-se a popularização do geoprocessamento. Novos recursos foram introduzidos nos SIG, em especial a estatística e o sensoriamento remoto (SR), o que contribuiu ainda mais para alavancar a sua popularização. Através de sensores orbitais (satélites) ou sensores aerotransportados (ex. aerofotos), tornou-se possível "descobrir", com maior rapidez e precisão, eventos ou fenômenos à distância. Tornou-se viável a captura de imagens de satélites de uma determinada localização em intervalos de tempo regulares e ajustáveis. Com esse recurso pode-se, por exemplo, monitorar queimadas e desmatamentos de forma mais efetiva e acurada. Outro uso dessa tecnologia é suprir informações de regiões onde os dados geográficos são insuficientes ou mesmo inexistentes. A união dos SIG com as ferramentas de sensoriamento remoto possibilitou o surgimento de novas aplicações, como o controle de determinadas endemias, causadas por fatores naturais e ambientais.

A estatística permite determinar a relação de dependências

1 A sigla SIG é normalmente utilizada tanto para representar o substantivo singular (sistema) quanto plural (sistemas).



entre variáveis indicando as relações de causa – efeito. Agregando então a geoestatística aos sistemas de geoprocessamento, variáveis geográficas são incluídas na

análise, ampliando assim as possibilidades de indicação das causas de determinado evento ou fenômeno. A comprovação de que o aumento de desmatamento em deter-

minadas regiões da Amazônia está associado à construção de estradas, ao gerar um aumento da ocupação humana, é um bom exemplo da aplicabilidade desses recursos.

## *Evolução dos SIG*

Os SIG surgiram na década de 60 no Canadá, como parte de um programa governamental para inventário e controle de recursos naturais. Esses SIG eram evidentemente limitados pela tecnologia da época: hospedados em grandes e onerosos computadores (*mainframes*), com terminais de vídeo de baixa resolução, funções reduzidas e quase nenhuma capacidade de análise. Eles eram, portanto, usados mais como substitutos da cartografia tradicional.

A partir de 1980 começam a aparecer os primeiros SIG comerciais, motivados basicamente pelo surgimento da microinformática, que promoveu a redução de custo de aquisição e manutenção de *hardware* e *software*. Pode-se citar ainda como outros fatores importantes: o desenvolvimento

tecnológico derivado diretamente da criação de diversos centros de pesquisa e desenvolvimento voltados para o geoprocessamento e a incorporação de novas ferramentas de análise, o que contribuiu ainda mais para a popularização desses produtos.

Mas, ainda nessa época, os SIG tinham uma capacidade limitada de incorporar dados alfanuméricos. Eles reconheciam apenas formatos de dados proprietários e apresentavam para seus usuários interfaces pouco amigáveis. Caracterizavam-se principalmente como ferramentas de automatização e representação cartográfica. Com o aumento do interesse na utilização da análise espacial, desenvolveram-se mais funções e iniciou-se a ligação entre os SIG e os sistemas de

gerenciamento de banco de dados (SGBD). Essa interligação possibilitou a manutenção dos dados alfanuméricos em SGBD, ampliando a utilização dos SIG.

O conceito de sistemas de geoinformação surgiu na década de 90 com a união entre os sistemas de informações tradicionais e os SIG. Para que isso ocorresse, foi necessário que a funcionalidade desses SIG fossem disponibilizadas através de interfaces (Application Programming Interface – API), acessíveis por outras ferramentas ou componentes de *software*. Essa tendência se ampliou ainda mais com a evolução dos SGBD, que passaram a suportar também o armazenamento e gerenciamento dos dados geográficos de uma aplicação, além dos seus dados alfanuméricos.

## *Arquiteturas de Gerência de Dados dos SIG*

Conforme apresentado em [1], atualmente um diferencial importante entre os SIG é a forma como os dados alfanuméricos e geográficos são armazenados e gerenciados. Há basicamente três arquiteturas de gerenciamento de dados

baseadas em SGBD: dual, integrada em um SGBD relacional, e integrada utilizando uma extensão espacial sobre um SGBD objeto-relacional. Embora não relacionada com o escopo deste artigo, existe ainda a arquitetura que utiliza

exclusivamente o sistema de arquivos do sistema operacional hospedeiro para armazenar tanto os dados alfanuméricos quanto as suas representações geográficas. Portanto, essa arquitetura não faz uso de nenhum SGBD.



## Arquitetura Dual

Nessa abordagem, os atributos ou dados alfanuméricos da aplicação são armazenados em tabelas (relações) de um banco de dados relacional, gerenciado por um SGBD. Os dados geográficos associados a esses atributos são armazenados isoladamente em arquivos do sistema operacional hospedeiro, em um formato proprietário do SIG. A associação lógica ou conceitual entre um dado geográfico e seu atributo alfanumérico (ex. população de uma cidade) é feita através do compartilhamento de um valor identificador único (chave primária) atribuído simultaneamente para ambos.

Essa é uma arquitetura simples e possibilita a utilização de qualquer SGBD do mercado, mesmo aqueles mais elementares, que não possuem nenhum suporte para o armazenamento e manipulação de objetos binários longos (BLOB). No entanto, o armazenamento dos dados geográficos fora

do banco de dados alfanumérico pode comprometer a sua qualidade e a eficiência do processo de consulta e de análise dos mesmos, pelas seguintes razões:

1. os dados geográficos estão fora do domínio de atuação dos mecanismos de *back-up*, recuperação, controle de transação e controle de concorrência do SGBD. Como consequência, fica comprometida a garantia de sua própria consistência e da integridade referencial em relação aos seus atributos, armazenados no banco de dados relacional;
2. a manipulação (consulta, inserção, alteração e remoção) dos dados tende a ser mais lenta, pois é efetuada em, geralmente, três etapas: uma pelo SGBD para a manipulação dos dados no banco de dados relacional, a outra pelo SIG – ou

componente de *software* à parte – sobre os arquivos que armazenam os dados geográficos e, finalmente, a terceira que irá consolidar, agrupar e relacionar os resultados obtidos nas etapas anteriores;

3. a integração com outras ferramentas (ex. outro SIG, *software* estatístico, etc.) torna-se mais complexa devido ao formato proprietário de armazenamento dos dados geográficos;
4. a administração do ambiente operacional torna-se mais complexa e onerosa, visto que usualmente as empresas delegam a profissionais com perfil e capacitação distintos a responsabilidade de administrar cada um desses ambientes: administrador de banco de dados e analista de suporte a sistemas operacionais.

## Arquitetura Integrada em um SGBD Relacional

Essa arquitetura considera que os dados geográficos serão armazenados em colunas do tipo objeto binário longo (BLOB) nas mesmas tabelas do banco de dados onde estão armazenados seus atributos alfanuméricos.

Embora elimine ou pelo menos reduza as desvantagens da Arquitetura Dual, essa abordagem ainda possui algumas restrições, que são inerentes à Teoria Relacional [2]. Por serem armazenados como cadeias binárias,

os dados geográficos não conseguem ter sua semântica interpretada pelo SGBD. Isso impossibilita a manipulação, via SQL<sup>2</sup> (Structured Query Language), dos dados geográficos baseada em seu conteúdo e na

2 Structured Query Language (SQL): linguagem de dados padrão dos SGBD relacionais do mercado e que engloba comandos para definição, inserção, consulta, atualização e remoção de dados, bem como a garantia de sua integridade e segurança de acesso.



associação desses dados aos seus atributos alfanuméricos. Cabe

então ainda ao SIG, ou outro componente de *software* a ele

associado, a tarefa de suprir essas deficiências.

## Arquitetura Integrada via Extensão Espacial

A arquitetura de gerência de dados dos SIG teve sua mais recente e principal evolução com a incorporação pelos SGBDs relacionais de recursos para a definição, armazenamento e manipulação de dados complexos através da orientação por objeto. Os produtos que seguiram essa tendência, tais como Oracle, DB2, Informix e PostgreSQL, passaram a ser denominados de SGBDs objeto-relacionais.

Com a capacidade de incorporar novos tipos de dados, inclusive objetos complexos, e de estender a linguagem de dados SQL com funções e operadores para manipulá-los, surgiram as extensões espaciais que agregam aos SGBD pelo menos as seguintes funcionalidades:

- suportam tipos de dados espaciais (tipo geométrico), tais como: ponto, linha, polígonos, etc;
- novas funções e operadores

foram incorporados à linguagem SQL para manipular dados espaciais, tais como: distância entre pontos, interseção entre linhas e polígonos. Além disso, é possível mesclar, em um mesmo comando SQL, operadores e predicados alfanuméricos e espaciais, o que possibilita a execução de consultas do tipo: "Quais são as cidades com mais de 100 mil habitantes que estão localizadas em um raio de 200 quilômetros de Belo Horizonte?";

- implementam novos mecanismos de indexação específicos para dados espaciais (R-Tree e Quad-Tree), que são devidamente incorporados ao processo de otimização de consultas do SGBD. Em conjunto, possuem efeitos extremamente positivos

no desempenho das consultas envolvendo dados espaciais, como aquela exemplificada anteriormente.

A partir de 2000, essa abordagem se consolidou com o estabelecimento de padrões abertos de serviços e intercâmbios de informações geográficas, determinadas pela Open Geospatial Consortium (OGC) [3]. A OGC é uma organização internacional sem fins lucrativos e formada por 271 associados, entre os quais: empresas, agências governamentais, universidades e institutos de pesquisa. Ela tem como objetivo o desenvolvimento de padrões consensuais para serviços geoespaciais e de localização. Cumprindo suas atribuições, esse consórcio estabeleceu padrões para a distribuição e interligações de dados de diversas fontes, facilitando a construção de Sistemas de Geoinformação [4].

## Comparando algumas Extensões Espaciais

Existem atualmente no mercado diversas opções, proprietárias ou livres, de extensão espacial para SGBD objeto-relacionais: IBM/DB2 Spatial Extender, Informix Spatial Datablade, MySQL Spatial Extension, Ora-

cle Spatial e PostgreSQL/Post GIS.

A Prodemge já utiliza alguns desses SGBD – DB2, MySQL e Oracle – mas sem fazer uso de seus mecanismos de extensão espacial. Levando em conta o interesse da empresa e de seus

clientes, no sentido de alavancar a utilização desse tipo de recurso nos sistemas existentes ou a serem desenvolvidos, foi realizado na Prodemge um estudo de prospecção de SGBD para geoprocessamento [5]. O seu



principal objetivo foi analisar e avaliar, sob o ponto de vista funcional, as características de alguns dos principais produtos do mercado, com o intuito de identificar aquele que possa ser melhor utilizado no desenvolvimento de sistemas de geoinformação no âmbito do Governo do Estado de Minas Gerais.

O escopo dessa avaliação ficou limitado aos produtos MySQL, Oracle e PostgreSQL, a fim de satisfazer as seguintes premissas:

1. excluir o DB2 do processo de análise, por ser um SGBD usado exclusivamente na Prodemge em aplicações legadas em *mainframe*;
2. restringir-se aos SGBD já utilizados na Prodemge para evitar uma proliferação excessiva de produtos na empresa e, assim, garantir o nível de qualidade de suporte técnico e reduzir os custos com treinamento dos funcionários;
3. para os produtos com licença de uso proprietária e paga, considerar aquele que possa ser disponibilizado em Linux, que é o sistema operacional com licença livre utilizado extensivamente na Prodemge e em seus clientes (vide premissa 1);
4. considerar apenas produtos com ampla divulgação e aceitação no mercado internacional e nacional.

A razão dessa premissa é minimizar os riscos de se optar por um produto cuja qualidade técnica não tenha sido devidamente comprovada por um universo significativo de usuários e, assim, consequentemente comprometer a qualidade de serviço da Prodemge.

Esse trabalho teve como principal balizador o documento OpenGIS Simple Features Specification For SQL [4] produzido pelo Open GIS Consortium (OGC). Esse documento, abreviadamente SFSSQL, é uma proposta de especificação de um conjunto de geometrias e de operações topológicas que geram novas geometrias em tipos de dados vetoriais. O seu principal objetivo, segundo seus realizadores, é definir um padrão da linguagem SQL que suporte o armazenamento, recuperação, consulta e atualização de dados geoespaciais, bem como de seus atributos descritivos alfanuméricos.

O escopo do trabalho se limitou à análise funcional dos produtos. Portanto, não foram realizados testes operacionais, sejam eles de instalação, avaliação de desempenho, de estabilidade e de facilidade de uso. A seguir estão brevemente descritos os produtos analisados, ressaltando algumas de suas características positivas e negativas. Uma visão mais detalhada está nas Tabelas de 1 a 5.

O MySQL [6] é um SGBD gratuito e de código aberto, podendo ser distribuído tanto com uma licença comercial quanto com uma integralmente baseada na GNU/GLP. É um produto muito utilizado e conhecido como sendo simples, fácil de operar e com excelente desempenho. A sua extensão espacial só foi disponibilizada a partir da versão 4.1.0, lançada em abril de 2003. A Prodemge vem utilizando-o nos últimos anos em sistemas *web* de pequeno e médio portes, porém sem fazer uso de seus recursos espaciais.

Embora sua implementação tenha seguido de perto as funcionalidades especificadas na SFSSQL, a sua extensão espacial apresenta algumas limitações que podem restringir significativamente a sua utilização, sendo que as principais são:

1. a extensão espacial só está disponível para o tipo de tabela MyISAM, que é a estrutura de armazenamento original e mais simples do MySQL. O problema é que, nesse caso, são inexistentes os mecanismos de controle de transação e de integridade referencial, o que limita a sua aplicabilidade a sistemas com baixo requisito de robustez e de complexidade. Essas limitações, no entanto, nem sempre são condizentes com a realidade da Prodemge e de seus clientes;



2. apenas um número muito reduzido de produtos da área de geoprocessamento faz interface atualmente com o MySQL (vide Tabela 5). A sua utilização fica mais uma vez restrita apenas ao desenvolvimento de sistemas autocontidos, sem interação com outros produtos ou componentes de *software* correlatos, tais como bibliotecas de *software*, servidores de exibição de mapas (*map servers*), exibidores de dados espaciais, etc.

O Oracle é um SGBD utilizado na Prodemge e em seus clientes desde meados da década de 90 e oferece dois tipos de recursos para manipulação de dados espaciais: o Oracle Locator e o Oracle Spatial. [7]

O Oracle Locator está incluído regularmente e sem custo adicional nas duas principais distribuições do Oracle: Standard Edition e Enterprise Edition. No entanto, ele provê um conjunto limitado de funcionalidades, que são úteis apenas em sistemas de geoinformação mais simples. De uma forma geral, ele oferece apenas um conjunto de recursos espaciais básicos e limitados, o que exclui o referenciamento linear (*linear referencing*), funções para análise espacial, transformações de coordenadas espaciais, entre outros.

O Oracle Spatial é uma extensão espacial só disponível na distribuição Oracle Enterprise Edition e representa uma solução tecnologicamente completa para

armazenamento, recuperação e análise de dados espaciais. Ela também adere à especificação SFSSQL, embora com total discordância em relação à nomenclatura original das funções e procedimentos. Esse fato, provavelmente por razões históricas e de compatibilidade, não só dificulta a identificação dessas funções e procedimentos perante a especificação original, como também limita a interoperabilidade com outras soluções baseadas na SFSSQL. Contraditório, portanto, com os próprios objetivos e premissas da padronização proposta pela OGC. Essa incoerência em relação aos padrões abertos não seria um obstáculo tão grande para a maioria dos seus usuários potenciais quanto é o valor de sua licença de uso – principalmente se comparado com a gratuidade dos outros dois produtos analisados. Além dos US\$ 40.000 por processador necessários para licenciar o Oracle Enterprise Edition, deve-se ainda desembolsar mais US\$ 10.000 por processador para habilitar a Oracle Spatial. O investimento inicial é então de no mínimo US\$ 50.000 mais os US\$ 11.000 por ano (22%) a título de manutenção (suporte técnico e atualização de versões).

O PostgreSQL [8] é um SGBD objeto-relacional, gratuito e distribuído sob a licença BSD. Ele teve sua origem no SGBD *Postgres*, que foi desenvolvido entre 1986 e 1993 na Universidade da Califórnia, em Berkeley. Em 1994, foi incorporado o suporte à linguagem SQL, visto que

curiosamente, até então, a sua linguagem de dados original era o PostQUEL, diferente da maioria dos produtos concorrentes que seguiram o padrão SQL. No mesmo ano, o seu código-fonte foi disponibilizado na *web*, onde um grupo de desenvolvedores mundialmente distribuídos o vem mantendo e atualizando. Apesar de não ser um produto tão popular como o MySQL, o PostgreSQL é bastante utilizado e reconhecido pela sua origem e qualidade acadêmica, bem como pela abrangência de sua funcionalidade [9].

Embora possua suporte nativo a dados espaciais, seus recursos originais não são muito abrangentes e eficientes. Para suprir essas deficiências e oferecer uma alternativa aos produtos com licença proprietária, a empresa Refrations Research Inc. (<http://www.refrations.net>) desenvolveu e mantém a extensão espacial PostGIS, que é distribuída gratuitamente sob licença GNU/GPL, e oferece ao PostgreSQL o suporte a objetos espaciais de acordo com a especificação SFSSQL. As funcionalidades dos recursos espaciais foram então ampliadas e diversos produtos de grande penetração e renome no mercado de geoprocessamento, com licença livre ou proprietária, possuem algum tipo de interface com essa solução. Concluindo, essas características o tornam, de acordo com o escopo e premissas do estudo efetuado, uma excelente alternativa técnica e econômica ao Oracle Enterprise Edition com a extensão Oracle Spatial.



Tabela 1: Identificação dos Produtos

	MySQL	Oracle Spatial	Oracle Locator	PostgreSQL/PostGIS
Fabricante	MySQL AB	Oracle Corporation	Oracle Corporation	PostgreSQL Global Development Group e Refrations Research
Versão analisada	4.1	10g	10g	8.0 (PostgreSQL) 0.9.1 (PostGIS)
Tipo de licença	GNU/GPL	Proprietária	Proprietária	PostgreSQL: BSD PostGIS: GNU/GPL
Custo do SGBD	Gratuito	US\$ 40.000/ processador	US\$ 5.000/ processador (Standard Edition One)	Gratuito
Custo da extensão espacial	Gratuito	US\$ 10.000/ processador	Incluído na licença do SGBD	Gratuito

Tabela 2: Requisitos de *Software*

	MySQL	Oracle Spatial	Oracle Locator	PostgreSQL/PostGIS
Requisitos de SGBD	<input type="radio"/>	Oracle Enterprise Edition	Oracle Standard Edition	Código-fonte do SGBD
Outros (bibliotecas de <i>software</i> , componentes, etc.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	GEOS, Proj4

- Funcionalidade ou característica: ATENDIDOS INTEGRALMENTE.
- ◉ Funcionalidade ou característica: ATENDIDOS PARCIALMENTE.
- Funcionalidade ou característica: NÃO ATENDIDOS.



Tabela 3: Compatibilidade com a SFSSQL

	MySQL	Oracle Spatial	Oracle Locator	PostgreSQL/PostGIS
Padrão SQL suportado pelas colunas geométricas	●	●	●	⊙
Formatos de dados espaciais suportados	●	⊙	⊙	●
Métodos (ou atributos) em geometrias	●	⊙	⊙	●
Métodos (ou atributos) para teste de relacionamentos espaciais	⊙	●	●	⊙
Métodos (ou atributos) de suporte à análise espacial	●	●	●	●
Métodos (ou atributos) sobre uma geometria ou coleção de geometrias	●	⊙	⊙	●
Tabelas de meta dados	○	●	●	●

- Funcionalidade ou característica: ATENDIDOS INTEGRALMENTE.
- ⊙ Funcionalidade ou característica: ATENDIDOS PARCIALMENTE.
- Funcionalidade ou característica: NÃO ATENDIDOS.



Tabela 4: Funcionalidade

	MySQL	Oracle Spatial	Oracle Locator	PostgreSQL/PostGIS
Tipos de objetos espaciais suportados	●	●	●	●
Número de dimensões suportadas nas geometrias	2	2, 3 e 4	2, 3 e 4	2, 3 e 4
Transformação de sistemas de coordenadas	○	●	○	●
Carga transacional de dados espaciais	●	●	●	●
Carga em lote de dados espaciais via utilitário proprietário	○	●	●	●
Suporte a dados geodésicos	○	●	●	○
Sistema de referência linear	○	●	○	○
Suporte ao armazenamento e manipulação de imagens <i>Raster</i>	○	●	○	○
Método de indexação espacial	<i>R-Tree (quadratic)</i>	<i>R-Tree e Quad-Tree</i>	<i>R-Tree</i>	<i>R-Tree (linear)</i>

● Funcionalidade ou característica: ATENDIDOS INTEGRALMENTE.

⊙ Funcionalidade ou característica: ATENDIDOS PARCIALMENTE.

○ Funcionalidade ou característica: NÃO ATENDIDOS.



Tabela 5: Conectividade com outros produtos

	MySQL	Oracle Spatial	Oracle Locator	PostgreSQL/PostGIS
ArcGIS	○	●	●	○
FME Engine	○	●	●	●
GeoServer	●	●	●	●
GeoTools	●	⊙	⊙	●
GDAL	○	●	●	●
GRASS	○	○	○	●
JUMP	○	○	○	●
MapInfo MapXtreme Java Ed.	○	⊙	⊙	○
MapInfo Professional	○	●	●	○
OGR Simple Feature Library	○	●	●	●
QGIS	●	○	○	●
TerraLib	●	●	●	●
UMN MapServer	○	●	●	●

- Funcionalidade ou característica: ATENDIDOS INTEGRALMENTE.
- ⊙ Funcionalidade ou característica: ATENDIDOS PARCIALMENTE.
- Funcionalidade ou característica: NÃO ATENDIDOS.

## Referências

- [1] Câmara, G. & Queiroz, G. R. *Arquitetura de Sistemas de Informação Geográfica*. In: Banco de Dados Geográficos. Capítulo em livro *on-line* disponível em abril/05 no endereço eletrônico: <http://www.dpi.inpe.br/gilberto/livro/bdados/>
- [2] Codd, E. *A Relational Model for Large Shared Data Banks*. CACM, 13:6, June 1970.
- [3] Open GIS Consortium, Inc. <http://www.opengeospatial.org/> (disponível em abril/ 05)
- [4] Open GIS Consortium, Inc. *OpenGIS Simple Features Specification For SQL Revision 1.1*. OpenGIS Project Document 99-049. May 5, 1999. Disponível em abril/05 no endereço eletrônico: <http://www.opengeospatial.org/docs/99-049.pdf>
- [5] Melo, C.H. *Prospecção de SGBD para Geoprocessamento*. Relatório Técnico Interno. Prodemge, fevereiro de 2005.
- [6] MySQL AB *MySQL Reference Manual - version 4.1*. Disponível em abril/05 no endereço eletrônico: <http://dev.mysql.com/doc/mysql/en/index.html>
- [7] Oracle Corporation. *Oracle Spatial - User's Guide and Reference 10g Release 1 (10.1)*. December, 2003.
- [8] PostgreSQL Global Development Group <http://www.postgresql.org> (disponível em abril/ 05)
- [9] Howard, P. *Pervasive joins the open source bandwagon*. IT-DIRECTOR.COM, 17th January, 2005. Disponível em abril/05 no endereço eletrônico: <http://it-director.com/article.php?articleid=12506>