

# Uso de Software *Open Source* nos SI de empresas

Miguel Andrade

[miguel.andrade@neoscopio.com](mailto:miguel.andrade@neoscopio.com)

## RESUMO

Neste trabalho pretende-se fazer uma introdução ao software *Open Source* orientada para o uso em sistemas de informação nas empresas. Após uma breve definição e enquadramento histórico, faz-se uma introdução explicativa do software segundo três vertentes: O modelo de desenvolvimento, a licença legal e o produto em si. Não se pretendeu ser exaustivo, preferindo-se destacar os seus aspectos mais relevantes do ponto de vista da empresa ou utilizador final. Conclui-se que existem grandes vantagens para a empresa em adoptar software *Open Source*, caso este exista, em alternativa a outro tipo de software, sendo estas vantagens nomeadamente alta qualidade, baixo custo, segurança, independência de fornecedor e grande adaptabilidade. São apresentadas sugestões para a escolha e avaliação deste tipo de software e dados exemplos de aplicações populares de algum do software *Open Source* existente.

## Keywords

Open Source, Sistemas de Informação, Tecnologias de Informação.

## INTRODUÇÃO

Software *Open Source* é software cuja licença de uso permite total liberdade de modificação e redistribuição do software, ou mesmo de modificações deste. Estas modificações são facilitadas dado que o software *Open Source* é obrigatoriamente distribuído com o código fonte que lhe deu origem (a *source*) e, na maior parte dos casos, é grátis. Dito de outra forma, o detentor de software deste género pode fazer as cópias que quiser do software vende-lo ou dá-lo a quem quiser sem que para isso tenha que pagar (ou pedir) qualquer licença ao fornecedor deste. Pode ainda modificar o software a seu bel-prazer, (pode por exemplo mudar-lhe o nome) e instala-lo em quantas máquinas quiser sem que tenha que pagar nada a ninguém por isso.

As licenças *Open Source* podem impor algumas restrições quanto à forma de redistribuição, mas para uma empresa ou consumidor final estes são os factos principais a ter em conta. Como este modelo de licenciamento de software está diametralmente oposto aos modelos de licenciamento proprietários mais populares, i.e. Microsoft, pode causar alguma perplexidade este tipo de software existir, quanto mais ser, em alguns casos, de qualidade superior aos seus equivalentes com políticas de licenciamento mais fechadas. Temos no entanto a evidencia nos últimos anos que este tipo de software se torna cada vez mais abundante e popular, tendo vindo a ser adoptado por grandes empresas como a IBM, HP, APPLE, SUN e NOVELL e um pouco por todo o mundo, desde pequenas empresas a governos de Países, sendo o sistema operativo GNU/LINUX uma expressão emblemática deste tipo de software.

Historicamente, este tipo de software (ou espírito de licenciamento) começou a ter expressão nos meios académicos, numa altura em que se desenvolviam as tecnologias que permitem hoje a existência da Internet. Muito software de apoio a estas tecnologias foi desenvolvido com este espírito, o que hoje se reflecte no facto de o melhor e mais popular software usado nestes domínios seja *Open Source*. Exemplos disto são: Apache<sup>1</sup> (Servidor de páginas de Internet), Sendmail<sup>2</sup> (Servidor de e-mail), Bind<sup>3</sup> (servidor que traduz endereços alfanuméricos nos seus equivalentes numéricos, i.e. [www.google.com](http://www.google.com) em 66.102.9.99), Perl<sup>4</sup> (linguagem muito popular para construção de páginas de Internet dinâmicas, tendo hoje vindo a ser substituído pelo PHP, também *Open Source*). Mais recentemente sistemas operativos *Open Source* (não apenas Linux, mas também BSD) são também usados em *sites* populares como o google, yahoo ou amazon.

Em 1985 um investigador do grupo de inteligência artificial do MIT, Richard M. Stallman<sup>5</sup> criou uma fundação, a FSF (Free Software Foundation)<sup>6</sup> que desde então se tem vindo a dedicar a apoiar e divulgar este tipo de software. A FSF não concorda muito com a nomenclatura *Open Source* preferindo (continuar a) chamar a este tipo de software, de *software livre*. Criou uma licença legal que consagra estes princípios de liberdade, a GPL<sup>7</sup> e desde a sua fundação, um dos seus mais ambiciosos projectos foi o de construir um sistema operativo de qualidade e totalmente *livre*, tendo optado por reproduzir o sistema operativo mais popular e promissor na altura, o UNIX. Chamou a este sistema operativo GNU (Abreviatura recursiva de GNU is Not Unix)<sup>8</sup>. Este projecto, de grande envergadura, nunca chegou a ser concluído totalmente pela FSF, dada a falta de um componente fundamental. O Kernel<sup>9</sup>.

Em 1990 um estudante da universidade de Helsínquia, Linus Torvalds, divulgou na Internet o seu projecto de construção de um Kernel compatível com Unix. Este projecto alcançou uma enorme popularidade entre vários programadores, que imediatamente se dispuseram a ajudar a esse projecto. Em muito pouco tempo, com a ajuda de programadores do mundo inteiro, foi construído um Kernel inicial fiável, que integrado com o software GNU já desenvolvido, constituía um sistema operativo completo. Linus chamou ao seu Kernel LINUX<sup>10</sup>, sendo hoje o sistema operativo obtido pela combinação GNU/LINUX mais conhecido apenas como LINUX.

Talvez mais importante que o Kernel em si, foi a maneira como este foi construído. Até então pensava-se impossível desenvolver um projecto da dimensão e complexidade do kernel através da Internet, usando a colaboração de um conjunto de programadores na sua maioria totalmente desconhecidos. O kernel provou que este modelo de desenvolvimento distribuído era válido e capaz de muito mais do que se poderia pensar. Este modelo de desenvolvimento foi seguido por muitos outros projectos de software, dando origem a uma autêntica revolução de projectos Open Source. O modelo foi estudado em grande detalhe por vários autores, sendo o estudo de Eric S. Raymond, “*the cathedral and the bazaar*”<sup>11</sup> publicado em 1992, o mais reconhecido e frequentemente apelidado de a “bíblia do software open source”. Os textos deste livro também podem ser encontrados *on-line* na página pessoal do autor em <http://catb.org/~esr>, onde foram publicados originalmente. Em 1998 Eric S. Raymond, e outros, cunharam o termo “Open Source” ([www.opensource.org](http://www.opensource.org)) tentando separar o software de quaisquer implicações moralistas ou filosóficas que a FSF publicitava para o software, até a altura conhecido apenas como *software livre*. Assim, de uma maneira geral *software livre* e software *Open Source* referem-se à mesma coisa, sendo

o último termo muito mais frequentemente usado hoje em dia, tendo também a vantagem de não causar confusões com a palavra “free” que em inglês significa igualmente “livre de custo”. A FSF faz grandes esforços para explicar que a palavra *free* em *free software* significa livre e não grátis (“*Free as in free speech not as in free beer*”). Embora a maior parte do Software Open Source tenha efectivamente um custo de aquisição nulo ou perto de zero, isto não é uma condição necessária, podendo este ser igualmente vendido.

## **SOFTWARE OPEN SOURCE**

Como forma a simplificar a análise do software *Open Source* com vista ao seu uso, vamos dividir esta em três partes; a análise do modelo, a análise da licença, e a análise do produto em si.

### **O modelo open source**

Diversos estudos<sup>12-15</sup> apontam para que o modelo de desenvolvimento de software Open Source conduz a uma melhor qualidade deste, nomeadamente maior fiabilidade, menos erros (*bugs*), melhor segurança e performance. Entre as várias razões apontadas, a mais referida é a de como o código dos programas é aberto e público, é revisto por vários programadores e técnicos quer por interesse dos próprios (ou das empresas que os empregam) quer por diversão. Erros são assim rapidamente detectados. O acesso ao código incentiva a que pessoas interessadas (além dos programadores originais) contribuam com sugestões e mesmo com alterações que completam e melhoram o programa original. Quando é detectada uma falha de segurança ela é rapidamente identificada e corrigida, ou pelos programadores originais, ou por qualquer outro.

Ainda sobre o modelo de desenvolvimento, interessa dizer que não existe um “dono” do software. É normal existir um conjunto de pessoas interessadas no seu desenvolvimento que “ocupam” a posição de dono e mantêm e coordenam o desenvolvimento do software. Contudo, qualquer pessoa (ou empresa) se pode propor a tal. Isto garante que desde que exista utilidade para o software, este dificilmente deixara de ser mantido por alguém. Por outro lado, leva a que possam existir versões semelhantes do mesmo software mantidas por pessoas ou grupos diferentes, mas isto é considerado porem como um aspecto positivo, apesar de aparentemente se estarem a desperdiçar recursos. Isto não significa que todo o software Open Source seja melhor que os seus equivalentes proprietários. Significa apenas que o modelo de desenvolvimento é considerado melhor. Se leva de facto a um melhor produto, depende de muitos outros factores, como por exemplo o número de potenciais interessados, uma vez que para o modelo funcionar em pleno, estes têm que ser em grande número.

## A licença

A licença é o que garante ao utilizador do software Open Source que o pode usar da maneira que bem entender. Traduz nela as liberdades que fazem com que software Open Source pertença ao modelo de desenvolvimento atrás descrito e com isso contenha em si as vantagens deste. A licença mais popular usada em software Open Source é a GPL<sup>7</sup> (General Public Licence) Publicada pela FSF. Uma tradução desta licença para português pode ser encontrada em [http://www.neoscopio.com/licenca\\_GPL\\_pt.txt](http://www.neoscopio.com/licenca_GPL_pt.txt). Outras licenças existem (em grande número) que consagram os mesmos princípios e que são compatíveis<sup>16</sup> com a GPL.

A licença do software tem vindo a assumir uma grande importância, uma vez que com a crescente popularidade do software Open Source tem vindo a aparecer muitos outros tipos de licenciamento de alguma maneira similares na forma mas não no conteúdo. Em particular, o facto de um determinado software disponibilizar o seu código fonte não o faz *Open Source*. A liberdade de livre redistribuição e modificação são essenciais para que o modelo atrás descrito funcione. É pois aconselhável que na dúvida se consulte uma lista de licenças compatíveis. Um exemplo de novas licenças é a Shared Source proposta pela Microsoft. Estas novas licenças tentam emular as vantagens do sistema Open Source mas mantêm a propriedade intelectual do software. Não permitem modificações nem redistribuição. Além disso o detentor dos direitos pode ainda processar alguém que, tendo visto a source de um dos seus productos, venha no futuro a trabalhar num producto semelhante, o que torna esta licença muito perigosa para qualquer programador.

Em resumo, as principais consequências da licença são:

- Liberdade de cópia.  
O software poderá ser copiado quantas vezes se quiser.
- Liberdade de uso.  
O Software poderá ser instalado em quantas máquinas se quiser.
- Liberdade de modificação.  
O Software poderá ser adaptado a necessidades específicas
- Liberdade de redistribuição.  
As adaptações feitas pela empresa ou por outros podem ser livremente dadas, trocadas ou vendidas.

Nenhuma destas liberdades está sujeita a qualquer pagamento adicional ou prévio.

## O produto

No final o que se obtém é uma cópia do software e uma cópia do seu código fonte. Para um utilizador final, o código fonte pode ser perfeitamente irrelevante e o software, como qualquer outro, pode ser bom ou mau. A licença é importante, mas só o é na prática se o software for bom. Aparecem diariamente vários projectos de software open source. Uma contagem num dos repositórios mais conhecidos ([www.sourceforge.net](http://www.sourceforge.net)) em 31 de

Dezembro de 2004 apresentava cerca de 93.000 Projectos. Poucos deles serão realmente excepcionais. Muitos provavelmente nunca chegaram sequer a ser úteis, mas muitos outros já são suficientemente bons para serem considerados e, cada vez mais frequentemente, são melhores que programas proprietários equivalentes. Isto é fácil de comprovar pelas estatísticas do seu uso. O facto de ter baixo custo ou mesmo nulo, não justifica *per se* a utilização de software, muito menos quando pensamos em empresas gigantes como a IBM.

Uma (extensa!) colectânea de estudos, estatísticas e relatórios sobre o uso de software Open Source foram coligidos e comentados por David A. Wheeler, em “*Why Open Source? Look at the numbers!*”<sup>17</sup>. Os documentos, na sua grande maioria produzidos por entidades de reputação reconhecida e mesmo de entidades governamentais de vários países e da própria União Europeia<sup>18</sup>, apontam no mesmo sentido: Open Source é (geralmente) melhor.

#### Promoção:

Consequência do modelo de desenvolvimento e da licença, o software Open Source não é normalmente alvo de promoção ou publicidade, outra que não a feita pelos seus próprios utilizadores. Igualmente, não tem normalmente associadas forças de venda, tornando-se assim muito diferente dos seus congéneres de licença proprietária: Não sendo promovido aos utilizadores finais, precisa de ser procurado por estes. Assim é necessário que departamentos de SI das empresas tenham elementos informados da existência de software Open Source, mas também as chefias necessitam de algum conhecimento deste género de software para apoiar as decisões de optar por ele. A dar resposta a estas necessidades, têm vindo a aparecer empresas de consultoria<sup>19</sup> que apoiam as empresas na escolha de tecnologias de informação que consideram as opções de software Open Source existentes.

#### Assistência:

A assistência a um software por parte de um fornecedor, ou pelo menos a possibilidade desta, é sempre importante, mas em certo tipo de software mais complexo, torna-se muitas vezes mais importante que o software em si. Um erro comum ao considerar software Open Source é o de pensar que este não tem assistência. Isto não é de todo verdadeiro. A Assistência torna-se a maior parte das vezes a principal fonte de receitas de quem escreve ou lidera projectos Open Source. Uma consequência da existência do código fonte é a de que qualquer pessoa ou empresa com suficientes conhecimentos técnicos pode prestar assistência a produtos Open Source. As empresas que fornecem consultoria de software Open Source fornecem igualmente serviços de assistência ao software recomendado. No caso da opção por software ser exclusivamente do utilizador final, terá de ser este, a procurar alguém que forneça assistência, pelas razões enunciadas. Mas, mesmo neste caso, encontra-se a assistência técnica procurada com bastante facilidade.

#### Custo:

A liberdade de modificação e de distribuição faz com que o custo de aquisição de software seja baixo ou mesmo nulo. Há no entanto que considerar outros custos, o que é hoje em dia popularmente conhecido como TCO (Total Cost of Ownership), ou seja, os custos totais de imputados ao uso de um certo software. Para esta análise vamos-nos concentrar apenas nas diferenças que podem existir entre software Open Source e software proprietário.

1. Custo inicial, ou de aquisição. Como vimos, uma das consequências do modelo de desenvolvimento e da licença, é o de os custos para software Open Source serem nulos ou reduzidos.
2. Custo de licenciamento. Estes são custos associados ao número de utilizadores que utiliza o software, ao número de máquinas em que este está instalado, ao número de máquinas que acedem a um software servidor, etc. Devido ao tipo de licenciamento, estes custos não existem para software Open Source.
3. Custo de assistência. Como vimos anteriormente, qualquer pessoa ou empresa com suficientes conhecimentos técnicos pode prestar serviços de assistência a software Open Source. A lei de concorrência num mercado livre faz com que estes custos sejam baixos e de alta qualidade.
4. Custos de hardware. O software Open Source é normalmente menos exigente em termos de hardware, pelo que se refere este ponto, onde se poderão também encontrar diferenças de custo.

#### **Vantagens. Conclusão.**

Caso exista software Open Source que satisfaça as necessidades de um utilizador ou empresa, a opção por este software tem, de uma forma geral, as seguintes vantagens:

Custo inicial baixo ou nulo. (consequência da liberdade de distribuição)

Custos de actualizações baixos ou nulos. (consequência da liberdade de distribuição)

Qualidade normalmente superior. (devido ao processo de desenvolvimento)

Integração com outros sistemas facilitada e adaptabilidade máxima. (devido à existência do código fonte)

Independência de fornecedor. (devido à existência do código fonte e da licença, o utilizador final pode sempre escolher um novo fornecedor de serviços ou fazê-lo ele próprio se desejar)

Maior garantia de continuidade (consequência do modelo de desenvolvimento e da independência de fornecedor)

Assistência de qualidade e de menor custo (consequência da existência do código fonte, da licença e da livre concorrência)

Menor TCO (custos totais suportados)

Livre de preocupações de auditorias. (há hoje em dia uma preocupação crescente nas empresas em utilizarem software para o qual não possuam licenças em número suficiente. Devido ao tipo de licenciamento do software Open Source não há razões para receios de auditorias a este tipo de software.

Maior Segurança. (devido ao modelo de desenvolvimento. Não só em relação a usos não autorizados, como também a vírus.)

## COMO ESCOLHER

A variedade nem sempre é uma coisa boa quando chega a altura de escolher. Por exemplo, uma pesquisa no sourceforge por “CRM” resulta em mais de 100 projectos. Há felizmente algumas técnicas que nos podem ajudar a determinar a qualidade de um projecto. Indicadores a ter em conta logo numa fase inicial são:

1. Indicadores de actividade (fornecidos pelos proprios repositórios)
2. Idade do projecto. Dois ou mais anos (com actividade) é um bom indicador de qualidade.
3. Data da última actualização. Esta deve ser proxima ( $\pm 6$  meses) da data da consulta.
4. Numero de mensagens nos foruns de apoio. Uma média superior a uma mensagem por dia desde o inicio do projecto é uma boa média.
5. Existencia de uma HomePage do projecto com conteudo actualizado.
6. Pesquisa em motores de busca pelo nome do projecto. Valores superiores a 5000 são bons. Valores superiores a 100.000 são possiveis. (Apache + “web server” = 5.6 milhões de resultados)
7. Pesquisa de “reviews” do software.

Estes indicadores devem ser entendidos como guia e não como regras, não dispensando por isso uma análise mais profunda do software.

Ainda assim a escolha não é muitas vezes fácil. Deve-se ter em conta no entanto que o custo de testar desistir e mudar de solução não é geralmente muito elevado.

## SOLUÇÕES OPEN SOURCE POPULARES

Simples ou de fácil acesso / integração.

Open Office. – Suite de aplicações semelhantes ao Microsoft Office. É compatível com este. Existe em português (como em outras línguas) e tem sensivelmente as mesmas funcionalidades. Pode ser fonte de grandes cortes em custos com software em PMEs e grandes empresas. [www.openoffice.org](http://www.openoffice.org)

Firefox – Alternativa ao Internet Explorer da Microsoft. É considerado muito melhor que este. Evita problemas com segurança e do chamado “addware”, publicidade indesejada. [www.getfirefox.com](http://www.getfirefox.com)

Thunderbird – Excelente cliente de e-mail. Evita largamente problemas com os chamados “worms” ou vírus que se transmitem por email, usando o cliente Outlook. [www.getthunderbird.com](http://www.getthunderbird.com)

Distribuições de Linux em LIVE-CD - Não necessitam de ser instaladas num disco. Correm a partir de um CD e, apesar de por isso estarem algo limitadas, dão uma boa ideia da usabilidade do sistema operativo Linux. <http://distrowatch.com/dwres.php?resource=cd>

Médias.

MySQL, PostgreSQL – Excelentes bases de dados. [www.mysql.com](http://www.mysql.com) , [www.postgresql.org](http://www.postgresql.org)

Apache Intranet Server. Plataformas de colaboração. [www.apache.org](http://www.apache.org)

PHP, interpretador de linguagem para aplicações de Internet. – Usado para fazer páginas de Internet com conteúdo dinâmico. [www.php.net](http://www.php.net)

Avançadas.

File Server (samba, Linux)<sup>17</sup> – Servidor de partilha de ficheiros. Pode substituir totalmente um servidor NT ou Windows 2000/2003 funcionando com clientes Linux ou Windows. [www.samba.org](http://www.samba.org)

Application Server (linux) – Permite que as aplicações (software) corram num servidor e assim poupar custos com hardware dos clientes que passam a ser simples terminais.

Mail server (sendmail, postfix) – permite ter o próprio servidor de email, podendo passar a exercer um controlo activo sobre este, como por exemplo para filtrar emails indesejados (spam). [www.postfix.org](http://www.postfix.org)

Proxy. (Squid) – Permite partilhar e controlar o acesso à Internet dentro de uma rede. [www.squid-cache.org](http://www.squid-cache.org)

Firewall – controla os acessos do exterior a uma rede interna e vice-versa. Protege as redes de acessos por pessoas não autorizadas. [www.smoothwall.org](http://www.smoothwall.org)

## REFERÊNCIAS E NOTAS

1. [www.apache.org](http://www.apache.org)
2. [www.sendmail.org](http://www.sendmail.org)
3. [www.isc.org/index.pl?sw/bind/](http://www.isc.org/index.pl?sw/bind/)
4. [www.perl.org](http://www.perl.org)
5. Também muito conhecido pelo seu login RMS (que são as suas iniciais).
6. [www.fsf.org](http://www.fsf.org)
7. General Public Licence. [www.fsf.org/licenses/licenses.html#GPL](http://www.fsf.org/licenses/licenses.html#GPL)
8. <http://www.fsf.org/philosophy/philosophy.html#AbouttheGNUproject>
9. Parte de um sistema operativo que implementa a interacção com o hardware e controla o acesso a recursos partilhados pelo sistema.
10. [www.linux.org](http://www.linux.org) [www.kernel.org](http://www.kernel.org)
11. Raymond, Eric S. (1999) *The Cathedral & the Bazaar*, O'Reilly, ISBN: 0-596-00108-8
12. (Linux), [www-106.ibm.com/developerworks/linux/library/l-rel/](http://www-106.ibm.com/developerworks/linux/library/l-rel/)
13. (MySQL), [www.reasoning.com/newsevents/pr/12\\_15\\_03.html](http://www.reasoning.com/newsevents/pr/12_15_03.html)
14. (Apache), [http://news.netcraft.com/archives/2004/06/02/most\\_reliable\\_hosting\\_providers\\_during\\_may.html](http://news.netcraft.com/archives/2004/06/02/most_reliable_hosting_providers_during_may.html)
15. (modelo de desenvolvimento), <http://opensource.mit.edu/papers/maccormackrusnakbaldwin.pdf>
16. <http://www.fsf.org/licenses/license-list.html#GPLCompatibleLicenses>
17. [http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)
18. <http://europa.eu.int/ida/oso>
19. O Autor é consultor nesta área.



Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Compartilhamento pela mesma licença. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-sa/2.0/br/> ou envie uma carta para Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.