

**UNIVERSIDADE FEDERAL FLUMINENSE  
FACULDADE DE ADMINISTRAÇÃO, CIÊNCIAS CONTÁBEIS E TURISMO  
PÓS-GRADUAÇÃO – LATO SENSU- MBA EM GESTÃO EMPRESARIAL  
E SISTEMAS DE INFORMAÇÕES**

**Denise Figueiredo Barbosa**

**ORIENTADOR: Prof. Newton Meyer Fleury (D.S.c)**

**Riscos e Obrigações envolvidas com o uso e a  
redistribuição de Open Source Software**



**Niterói**

**2012**



**Universidade Federal Fluminense**

**Faculdade de Administração, Ciências Contábeis e Turismo.**

**Pós-graduação – Lato Sensu - MBA em Gestão Empresarial e Sistemas de  
Informações**

**Denise Figueiredo Barbosa**

**Riscos e Obrigações envolvidas com o uso e a redistribuição de  
Open Source Software**

**Niterói**

**2012**

Denise Figueiredo Barbosa

**Riscos e Obrigações envolvidas com o uso e a redistribuição de  
Open Source Software**

Artigo apresentado ao Curso de  
Pós-Graduação Latu Sensu MBA de  
Gestão Empresarial e Sistemas de Informações  
da Universidade Federal Fluminense  
como requisito parcial para obtenção do  
certificado de conclusão.

ORIENTADOR: Prof. Newton Meyer Fleury (D.S.c)

Niterói

Maio/2012

## Resumo

É cada vez maior o número de empresas que usam e redistribuem softwares de “código aberto” em seus produtos e serviços, são os chamados OSS, sigla para o termo inglês “*Open Source Software*”. Porém, poucas estão cientes das obrigações e riscos envolvidos pelo não cumprimento das obrigações legais, a que estão sujeitas quando utilizam, modificam, integram ou redistribuem um OSS. O uso correto, do ponto de vista legal, de um OSS é, portanto, um assunto complexo que precisa levar em consideração as diferentes formas de licenciamento.

Esse artigo se propõe a apresentar, de forma simples, uma síntese do conteúdo diversificado de informações coletadas em artigos acadêmicos, sites na internet e publicações ligadas ao tema. Seu principal objetivo é apresentar as especificidades do licenciamento dos OSS, além de um trazer um breve histórico do movimento “*Open Source*” e de realizar uma análise da sua dinâmica. Também estão relacionadas e descritas as características das principais licenças OSS, com foco nas obrigações e riscos envolvidos na sua integração e distribuição.

**Palavras-Chave:** Código Aberto, Software Livre, Software Proprietário e Direito Autoral.

## Abstract

It is increasing the number of companies that are using and redistributing Open Source Software (OSS) in their products and services, but just few of them are aware of the obligations and risks involved, when do not fulfill the legal requirements, that all of them are subjected, when using, modifying, integrating or redistributing an OSS. The correct use of them, in the legal point of view, is, therefore, a complex issue that needs to take into account the different forms of licensing.

This article aims to present, in a simple way, a synthesis of the diverse content of information, which were collected in academic articles, web sites and professional publications related to the subject. The main purpose is to present the specificities of the OSS licensing and a brief historical of the Open Source Movement, besides to perform an analysis of their dynamic. Also are listed and described the main characteristics of OSS licenses, focusing on the legal obligations and involved risk in their integration and distribution.

**Key-words:** *Open Source Software, Copyleft, Free Software, Proprietary Software and Copyright.*

# Sumário

Lista de Tabelas	viii
Lista de Figuras	ix
Lista de Siglas	x
1. Introdução	1
2. Software de Código Aberto e Software Livre	2
2.1. Definições	2
2.2. O Movimento Open Source	3
2.2.1. Definição de Software Livre	3
2.2.2. Definição de Código Aberto	5
2.2.3. Erros de entendimento comuns referentes aos termos: Software Livre e Código Aberto	7
3. Licenças de Software Livre e Licenças de Software de Código Aberto.	10
3.1. Licenças Permissivas	11
3.1.1. Licença BSD / Berkeley Software Distribution	11
3.1.2. Licença MIT/X11	13
3.1.3. Licença Apache	14
3.1.4. Outras Licenças Permissivas	15
3.2. Licenças Recíprocas	15
3.2.1. Licenças Recíprocas Totais	15
3.2.1.1. Licença GNU GPL (General Public License)	15
3.2.1.2. Licença AGPL (GNU Affero General Public License)	17
3.2.2. Licenças Recíprocas Parciais	18
3.2.2.1. Licença LGPL (GNU Lesser General Public License)	19
3.2.2.2. Licença Mozilla/MPL (Mozilla Public License)	19
3.2.2.3. Licença EPL (Eclipse Public license)	20
3.3. Comparativo das principais licenças de Software Livre e/ou OSS	21
3.4. Metodologia para Escolha de Licenças em um Projeto de Software Livre	22
3.5. Comparativo entre as principais licenças de software livre e de código aberto	23
4. Obrigações e Riscos Envolvidos	24
4.1. Entendendo o Mecanismo de “Copyleft” Open Source Licenses	25
4.2. Obrigações impostas ao licenciado de OSS.	26
4.3. Riscos envolvidos com o descumprimento dos Termos e Condições de Licenciamento OSS	27

4.4. Programas derivados de OSS	29
5. Conclusão	31
Bibliografia	33

## **Lista de Tabelas**

Tabela 1 – Quadro Comparativo das Principais Licenças de Software Livre e OSS.....	21
Tabela 2 - Tabela comparativa das práticas de licenciamento entre software livre e código aberto.....	24



# Lista de Figuras

Figura 1 - Logotipo GNU.....	3
Figura 2 - Open Source TM .....	7
Figura 3 - Softwares Open Source .....	9
Figura 4 - Closed Source TM.....	12
Figura 5 - Metodologia para escolha de licenças .....	23
Figura 6 - Copyright e Copyleft.....	25

## **Lista de Siglas**

**OSS** – Open Source Software

**OSI** – Open Source Initiative

**OSD** – Open Source Definition

**FSF** – Free Software Foundation

**MIT** – Massachusetts Institute of Technology

**BSD** – Berkeley Software Distribution

**GPL** – General Public License

**LGPL** – Lesser General Public License

**DFSG** – Debian Free Software Guidelines

**MPL** – Mozilla Public License

**CPL** – Common Public License

# 1. Introdução

Este artigo pretende realizar um levantamento das principais formas de licenciamento do chamado Softwares de Código Aberto (OSS, *Open Source Software*<sup>1</sup>). Os diversos tipos de licenças serão apresentados aqui de forma simples, de modo a assegurar que seus conceitos fundamentais serão facilmente compreendidos, inclusive por aqueles que não possuem um conhecimento prévio do assunto. O objetivo é informar as principais vantagens e desvantagens de cada modelo, bem como as consequências e riscos envolvidos quando se utiliza OSS.

É sabido que, os chamados OSS podem, em alguns casos, ser modificados e/ou integrados com os tradicionais softwares proprietários, resultando em uma nova solução com diferentes aplicabilidades. Dois exemplos muito comuns, encontrados nestes ambientes mistos de desenvolvimento, e que serão objetos deste estudo, são:

(1) um software proprietário, desenvolvido por uma empresa com um fim específico, utiliza elementos ou interage com um programa OSS e, por este motivo, este programa OSS acaba sendo redistribuído junto com o software proprietário;

(2) um software proprietário utiliza alguma biblioteca padrão regulamentada e distribuída sob uma licença de código aberto.

O estudo iniciará com uma breve definição do termo “Código Aberto”, seguido de um breve histórico do movimento *Open Source*, apresentando suas principais características, bem como um interessante estudo sobre os aspectos legais que não podem ser desconsiderados quando se analisa este tema. Outro ponto importante que será abordado é a compatibilidade entre as diversas formas de licenciamento, fator fundamental nos processos de integração de diferentes componentes, dentro da realidade dos nossos ambientes de desenvolvimento, que muitas vezes mesclam softwares proprietários e programas OSS.

Também serão abordados interessantes conceitos deste mundo *Open Source*. Termos como *Copyleft* e *Free*; conceitos como direitos autorais e propriedade intelectual; as diretrizes da *Open Source Initiative* (OSI)<sup>2</sup> e da *Free Software Foundation* (FSF)<sup>3</sup>; as diferenças entre os movimentos "Software Livre" e "Código Aberto", entre outros.

Todos esses conceitos serão apresentados com um único objetivo: reunir informações relevantes que nos permitam responder a pergunta que inquieta os diretamente envolvidos com o tema e já conscientes de sua complexidade: “*Afinal que obrigações legais precisamos*

---

<sup>1</sup> O termo **código aberto**, ou *open source* em inglês, foi criado pela *OSI* (*Open Source Initiative*) e refere-se a *software* também conhecido por *software livre*. Fonte: [http://pt.wikipedia.org/wiki/C%C3%B3digo\\_aberto](http://pt.wikipedia.org/wiki/C%C3%B3digo_aberto)

<sup>2</sup> A **Open Source Initiative** (OSI) - Iniciativa pelo código aberto - é uma organização dedicada a promover o software de código aberto ou software livre.

<sup>3</sup> A **Free Software Foundation** (FSF, *Fundação para o Software Livre*) é uma organização sem fins lucrativos e que se dedica a eliminação de restrições sobre a cópia, redistribuição, estudo e modificação de programas de computadores – bandeiras do movimento do software livre, em essência. Fonte: [http://pt.wikipedia.org/wiki/Free\\_Software\\_Foundation](http://pt.wikipedia.org/wiki/Free_Software_Foundation)

*cumprir quando utilizamos, modificamos, integramos e redistribuímos Software de Código Aberto?”.*

Espero com isso, colaborar com todos aqueles que não são especialistas em licenciamento de software, mas que precisam entender sua complexidade em função dos papéis que desempenham dentro das suas organizações. Tornar compreensíveis os principais elementos das regras pertinentes ao tema, visando facilitar o processo decisório, que envolve a aplicabilidade dos OSS e suas diferentes formas de licenciamento, é essencial para responder a pergunta-base que orienta toda essa pesquisa.

## 2. Software de Código Aberto e Software Livre

### 2.1. Definições

[...] *“software livre” é uma questão de liberdade, não de preço. Para entender o conceito, você deve pensar em liberdade no sentido de “liberdade de expressão”, não de “gratuidade”.* (Fonte: [www.gnu.org](http://www.gnu.org)).

Não é novidade que os softwares, de um modo geral, contam com a proteção de direitos autorais em quase todos os países do mundo. Normalmente o Direito Autoral é utilizado por um autor para proibir outros de usar, copiar, modificar ou distribuir seu trabalho sem a sua permissão. Além disso, no caso de um Software Proprietário, na maioria dos casos, apenas o código objeto ou executável é disponibilizado para os usuários e, portanto, muito dificilmente oferecem alguma possibilidade de adaptação. Em contraste com esse modelo, uma licença OSS prevê que todos os que recebem uma cópia do software, tem permissão para usá-lo livremente, copiá-lo, modificá-lo ou mesmo redistribuí-lo. Para permitir que o destinatário do OSS possa adaptá-lo para os seus propósitos e desenvolvê-lo ainda mais, os autores de Softwares Open Source sempre fornecem o código fonte dos seus programas. Afinal essa é uma regra primordial do mundo Open Source.

[...] *“software livre” é uma questão de liberdade dos usuários. Liberdade para executar, copiar, distribuir, estudar, mudar e melhorar o software.* (Fonte: [www.gnu.org](http://www.gnu.org)).

É bastante comum ver os termos Código Aberto (*Open Source*) e Software Livre (*Software Free*) sendo tratados como sinônimos. Da mesma forma que não é raro encontrar a expressão “Código Aberto” como mero sinônimo de “Código-Fonte Aberto”. Não há, necessariamente, erros nesse entendimento, mas é importante estar ciente de que há diferenças significativas entre estes dois mundos. Vamos entendê-las:

Basicamente, qualquer licença de *software livre (Free Software)* é também uma licença de código aberto (*Open Source*), a diferença entre as duas nomenclaturas reside

essencialmente na sua apresentação e em algumas discussões conceituais. As duas principais organizações internacionais responsáveis pela proteção e promoção do software livre, a *Free Software Foundation* (FSF) e a *Open Source Initiative* (OSI), atuam para garantir que os termos *Free Software* e *Open Source* sejam utilizados de forma correta.

Enquanto a *Free Software Foundation*, uma organização sem fins lucrativos que se dedica a eliminação de restrições sobre a cópia, a redistribuição, o estudo e a modificação de programas de computador, usa o termo "Software Livre" envolto de um discurso baseado em questões éticas, de direitos e liberdades essenciais, a *Open Source Initiative*, outra organização dedicada a promover o software *Open Source*, usa o termo "Código Aberto" ou "Software Livre" sob um ponto de vista puramente técnico. Mas é interessante observar que ambas as organizações nasceram a partir da mesma história do Unix, do software livre e da cultura hacker. Veremos no decorrer deste artigo as principais diferenças entre esses dois movimentos, visando entender seus conceitos e valores.

## 2.2. O Movimento Open Source

### 2.2.1. Definição de Software Livre

[...] "*free speech, not free beer*" em português: "*liberdade de expressão, não cerveja de graça*". (Fonte: [www.gnu.org](http://www.gnu.org)).



Figura 1 - Logotipo GNU

O movimento *Free Software* teve início em 1984, quando Richard Matthew Stallman, um cientista da computação e famoso hacker americano, deu início ao *Projeto GNU*

com o objetivo de criar um sistema operacional totalmente livre, o GNU. Stallman desejava desenvolver um "sistema de software completamente compatível com o Unix, mas que não utilizasse o código fonte deste", surgiu aí o nome GNU, um acrônimo recursivo de: “*GNU is Not Unix*” (em português: “GNU Não é Unix”). Esse movimento culminou com a criação, em Outubro de 1985, da *Free Software Foundation* que, desde então, dedica-se a eliminação de restrições sobre a cópia, redistribuição, estudo e modificação de programas de computadores – bandeiras do movimento do software livre, em essência.

Em 1991, o movimento *Software Livre* ganha ainda mais projeção, quando um jovem finlandês chamado Linus Torvalds criou um núcleo de sistema que poderia usar todas as peças do sistema operacional GNU. Este núcleo ficou mundialmente conhecido como Linux<sup>4</sup>, uma contração de Linus e Unix.

A *Free Software Foundation* considera um software como livre quando atende aos quatro tipos de liberdades essenciais para os seus usuários:

- Liberdade para executar o programa, para qualquer propósito;
- Liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades;
- Liberdade de redistribuir cópias do programa de modo que você possa ajudar ao seu próximo;
- Liberdade de modificar o programa e distribuir estas modificações, de modo que toda a comunidade se beneficie.

Isso equivale a dizer que, segundo a *FSF*, um programa é “software livre” se os seus usuários possuem essas 04 (quatro) liberdades asseguradas. Resumidamente: “Liberdade para executar, estudar, adaptar, redistribuir cópias modificadas ou não, gratuitamente ou cobrando uma taxa pela distribuição, a qualquer um, em qualquer lugar”. E eles acrescentam:

[...] *Ser livre para fazer tudo isso significa (entre outras coisas) que você não deve ter que pedir ou pagar pela permissão de fazê-lo.* (Fonte: [www.gnu.org](http://www.gnu.org)).

É princípio fundamental, para que essas quatro liberdades sejam satisfeitas, que o software seja sempre distribuído juntamente com o seu código-fonte e que não seja colocada nenhuma restrição para que os usuários alterem e/ou redistribuam esse código.

“Software Livre” é, sem dúvida, um conceito extremamente difundido no moderno mundo da computação. A forte relação entre ele e os aspectos de liberdade preconizados pela *Free Software Foundation* faz com que o termo seja utilizado para

---

<sup>4</sup> **Linux** é um termo popularmente utilizado para se referir a sistemas operacionais que utilizem o núcleo Linux. O seu código fonte está disponível sob a licença GPL (versão 2) para que qualquer pessoa o possa utilizar, estudar, modificar e distribuir livremente de acordo com os termos da licença. Fonte: <http://pt.wikipedia.org/wiki/Linux>

designar muito mais que uma categoria de software, ele representa, na visão de seus membros, “um movimento social, que defende uma causa”.

### 2.2.2. Definição de Código Aberto

[...] “*Código Aberto*” não significa apenas acesso ao código-fonte. Os termos de distribuição de software *Open Source* devem obedecer aos 10 critérios listados na *Open Source Definition*. (Fonte: [www.opensource.org](http://www.opensource.org)).

A *Open Source Initiative* - Iniciativa pelo código aberto - é uma organização dedicada a promover o chamado software de “Código Aberto”. Ela foi criada com o objetivo de incentivar uma aproximação maior das entidades comerciais com o então chamado “Software Livre”. Sua atuação principal é a de certificar quais licenças se enquadram como licenças de software livre, além de promover e divulgar o software *Open Source* e de suas vantagens tecnológicas e econômicas.

A organização foi fundada em fevereiro de 1998, por Bruce Perens e por Eric S. Raymond. A formação da OSI começou com a publicação de trabalho de Raymond, intitulado “A Catedral e o Bazar” em 1997; Raymond apresentou este trabalho na *O’reilly Perl Conference*, o que, na ocasião, ajudou a desencadear a decisão da Netscape em liberar o código-fonte, como software livre, do mais popular navegador da Web na ocasião, isso aconteceu em 22 de janeiro de 1998.

A OSI, assim como muitos membros desta comunidade, considera que o software é, em primeiro lugar, uma ferramenta, e que o mérito dessa ferramenta deve ser julgado com base em critérios puramente técnicos. Para eles, o software de código aberto é, no longo prazo, economicamente mais eficiente e de melhor qualidade e, por isso mesmo, deve ser incentivado.

Dentre as primeiras entidades que se registraram como membros da *Open Source Initiative* temos: Apache Software Foundation, FreeBSD, KDE, Linux Foundation, Mozilla Foundation, entre outras.

A *Open Source Initiative* optou pelo uso do termo *open source* ao invés de *free software* para, nas palavras de Michael Tiemann (seu atual presidente):

[...] *Se livrar da atitude moralizadora e de confronto que tinha sido associada anteriormente ao termo software livre.*

Fonte: [http://pt.wikipedia.org/wiki/Open\\_Source\\_Initiative](http://pt.wikipedia.org/wiki/Open_Source_Initiative).

Para o OSI, a disponibilidade do código-fonte não é condição suficiente para que ele seja considerado software de Código Aberto, ela entende que é necessário satisfazer

10 (dez) critérios (listados abaixo), inspirados nas “Orientações sobre Software Livre” do projeto Debian - um projeto oficialmente fundado por Ian Murdock em 16 de Agosto de 1993, que reúne um grupo mundial de voluntários que se esforçam para produzir um sistema operacional livre, composto inteiramente por software livre. O produto principal do projeto é a distribuição Debian GNU/Linux, que inclui o kernel do sistema operacional Linux e centenas de aplicações pré-empacotadas. (Fonte: <http://www.debian.org>).

### ***Open Source Definition (OSD)***

Fonte: [www.opensource.org/docs/OSD](http://www.opensource.org/docs/OSD)

[...] *Open Source Definition* é um documento publicado pela OSI para determinar se uma licença de software pode ser rotulada com a marca da certificação Open Source.

(Fonte: [http://en.wikipedia.org/wiki/The\\_Open\\_Source\\_Definition](http://en.wikipedia.org/wiki/The_Open_Source_Definition)).

Os termos de distribuição de software open-source devem obedecer aos seguintes critérios:

- 1. Livre redistribuição:** A licença não deve restringir ninguém de vender ou doar o software para terceiros, como um componente de uma distribuição agregada de software contendo programas de várias fontes diferentes. A licença não deve exigir royalty ou outra taxa para tal venda.
- 2. Código-fonte:** O programa deve, obrigatoriamente, incluir o código fonte e deve permitir a distribuição do código fonte, bem como do formato compilado. Quando alguma forma de um produto não for distribuída com o código fonte deve, então, haver um meio bem divulgado de obter o código fonte (oferecendo, por exemplo, download através da Internet, sem custo). O código fonte deve ser a forma preferida na qual um programador irá alterar o programa e o mesmo não pode ser deliberadamente ofuscado.
- 3. Obras derivadas:** A licença deve permitir modificações e derivações que possam ser redistribuídas dentro dos mesmos termos da licença original;
- 4. Integridade do código do autor:** A licença pode restringir a distribuição do código fonte em forma modificada, mas somente se a licença permitir a distribuição de "*patch files*" com o código fonte, com o propósito de modificar o programa em tempo de compilação. A licença deve permitir, explicitamente, a distribuição do software construído a partir do código fonte modificado. A licença pode exigir que, as obras derivadas tenham nome ou número de versão diferente do software original.



5. **Não discriminar pessoas ou grupos:** A licença não pode, de forma alguma, discriminar nenhuma pessoa ou grupos de pessoas;
6. **Não discriminar nenhuma área de utilização:** A licença não deve restringir ninguém de fazer uso do programa em um campo específico de atuação. Por exemplo, ela não pode restringir o programa de ser usado em uma empresa, ou de ser usado para uma pesquisa genética.
7. **Distribuição da licença:** Os direitos associados ao programa através da licença são automaticamente repassados a todas as pessoas às quais o programa é redistribuído, sem a necessidade de definição ou aceitação de uma nova licença por aquelas pessoas;
8. **A Licença não pode ser específica de um produto:** Os direitos associados a um programa não dependem de qual distribuição em particular aquele programa está inserido. Se o programa é extraído de uma distribuição, e usado ou distribuído dentro dos termos da licença do programa, os direitos garantidos por sua licença continuam valendo;
9. **Licenças não podem restringir outro software:** A licença não pode colocar restrições em relação a outros programas que sejam distribuídos junto com o software em questão; Por exemplo, a licença não deve insistir que outros programas distribuídos, na mesma mídia, devam ser *open-source* software.
10. **Licenças devem ser neutras em relação às tecnologias:** Nenhuma exigência da licença pode ser específica a uma determinada tecnologia ou estilo de interface.



Figura 2 - Open Source TM

### 2.2.3. Erros de entendimento comuns referentes aos termos: Software Livre e Código Aberto

Conhecidos os critérios, vale observar a análise de Stallman (FSF) para a discussão das diferenças entre Software Livre e Código Aberto; ele afirma que “a diferença é

filosófica”. E ainda descreve a FSF e a OSI como “organizações separadas dentro da mesma comunidade de software livre”. Ainda de acordo com Stallman:

[...] *A Foundation discorda da OSI em seus objetivos e valores básicos, mas os seus pontos de vista e os nossos conduzem, em muitos casos, para o mesmo comportamento prático - o desenvolvimento do software livre. Como resultado, as pessoas do movimento software livre e do movimento código aberto muitas vezes trabalham em conjunto em projetos práticos de desenvolvimento de software.*

(Fonte: [http://pt.wikipedia.org/wiki/Open\\_Source\\_Initiative](http://pt.wikipedia.org/wiki/Open_Source_Initiative)).

Porém, conceitualmente, parece haver uma disputa para se determinar qual termo remete mais facilmente à ideia, conceitos e valores defendidos por ambas as comunidades.

A *Free Software Foundation* já declarou estar ciente que o termo Software Livre é propenso a erros de interpretação: um significado não intencional que leva a crer ser *free* todo software “*que você pode obter de graça, a preço zero*”; ela acredita minimizar o problema com a definição marcante de que *free* é o software “*que dá ao usuário certas liberdades*”, mas concorda que o ideal seria “*a identificação de um termo menos ambíguo*”.

A situação pareça mais crítica para a comunidade *Open Source*, o significado, aparentemente óbvio, para a expressão *Open Source Software* leva a maioria das pessoas a pensar que isso significa apenas que “você pode olhar o código fonte”. Esse entendimento é ainda mais simplório do que o entendimento geral para o termo Software Livre, e não está absolutamente alinhado com a definição oficial da OSI. Acreditar que OSS é simplesmente “software que você pode olhar o código fonte” fica muito aquém do seu real significado e acaba incluindo programas que não são nem “free” nem “open source”.

A comunidade OSI tentar lidar com isso, apontando para a sua definição oficial, mas a abordagem corretiva é menos eficaz para eles do que é para a FSF. O termo Software Livre tem dois significados naturais, um dos quais é o significado pretendido. A FSF defende que se uma pessoa compreendeu a ideia da frase emblemática - “*é liberdade de expressão, não cerveja de graça*” – ela não vai mais errar o entendimento. Já o termo *Open Source* tem apenas um significado natural, e que é bem diferente do significado que sua comunidade pretende.

Outro equívoco muito comum associado ao Open Source Software é a ideia de que “*sendo OSS significa que não usa a GNU-GPL*”. Isso tende a acompanhar outro erro de entendimento: a ideia de que Software Livre é todo software “*coberto somente pela GNU-GPL*”. Ambos estão errados, afinal a licença de Software Livre GNU-GPL (que veremos mais adiante) qualifica, sim, uma licença Open Source e, da mesma forma, a maioria das licenças Open Source qualifica, sim, Software Livre.

Resumidamente, podemos considerar que a diferença primordial entre o movimento Open Source (código aberto) e o Software Livre, está realmente no plano dos valores. Para o movimento Open Source, o fato de um software ser aberto é uma questão de praticidade, não de ética. Para a comunidade Software Livre, a questão é fundamentalmente social. As diferenças entre os dois movimentos podem ser mais bem entendidas compreendendo a diferença entre as licenças BSD (*Open Source*) e as licenças GPL (*Free Software*). É importante compreender que, para ambas as comunidades ser *Open Source* e ver o código fonte (ou parte dele) são coisas muito diferentes. Da mesma forma que ser *Free Software* e ser de graça é bem diferente.



Figura 3 - Softwares Open Source

### 3. Licenças de Software Livre e Licenças de Software de Código Aberto.

É fácil concluir que o simples fato de um programa ter seu código fonte disponível ou aberto não garante absolutamente nada sobre a sua distribuição, modificação e comercialização. Na verdade, é o desenvolvedor do programa quem determina suas condições de uso. O detentor dos direitos sobre um software, quando decide torná-lo livre, deve escolher os termos em que seu programa será distribuído, ou seja, quais direitos ele estará transferindo para as outras pessoas e em que condições. O documento que formaliza esse ato é denominado “Licença de Software Livre” ou “Licença de Software de Código Aberto” e, normalmente, é distribuído junto com o código fonte.

São inúmeras as possibilidades para redigir o texto de uma licença de Software Livre ou de Código Aberto, mas a prática mais comum é reaproveitar alguma das licenças já consolidadas nessas comunidades. Com isso, reduz-se a proliferação de licenças, o que deve realmente ser evitado, pois gera trabalho adicional para os usuários, uma vez que se torna necessário estudar os termos de cada nova licença presente no software que será utilizado.

O portal *Sourceforge* (<http://sourceforge.net>), maior portal repositório de projetos open source disponível atualmente – oferece algumas opções de licenças para quem quer criar um projeto *Open Source*, que representam as mais comuns e populares licenças OSS em uso:

- GNU General Public License (GPL)
- GNU Library or Lesser General Public License (LGPL)
- BSD License
- Apache License
- MIT License
- Mozilla Public License

As licenças de software livre permitem que o desenvolvedor do software abdique de alguns direitos seus, estabelecidos pelo *Copyright (Direitos Autorais)* – conceito jurídico, promulgado pela maioria dos governos, dando o criador de um trabalho original, direitos exclusivos sobre ele, geralmente por um tempo limitado – determinando quais são esses direitos e suas condições de aplicação.

A popularização das licenças de software livre ou de código aberto se deve, principalmente, a garantia jurídica que estas licenças fornecem aos seus utilizadores, assegurando que estes não estão cometendo nenhum ato de infração de direito autoral (*Copyright*) ao copiar ou modificar o software.

Basicamente, as licenças de código aberto e/ou software livre podem ser separadas em tres (03) categorias - divididas de acordo com a presença de termos que impõem restrições de licenciamento na redistribuição ou criação de trabalhos derivados do original - a saber: Licenças Permissivas, Recíprocas totais e Recíprocas parciais.

O estudo a seguir tomou como base diversas informações coletadas em alguns dos principais sítios da internet que concentram importantes artigos sobre o mundo Open Source, tais como, o OpenSource.org, Apache.org, GNU.org, Wikipedia, FSF.org, entre outros.

### **3.1. Licenças Permissivas**

Segundo a Wikipedia, as licenças permissivas recebem este nome pelas poucas restrições que impõem às pessoas que obtém o produto. Estas licenças devem ser usadas quando se deseja que o projeto possa atingir um número maior de pessoas, para ampla divulgação. Em geral, permitem que qualquer detentor do código fonte possa desenvolver um produto derivado e fechar o código para explorá-lo comercialmente (se assim o desejar).

*[...] e tem as permissivas que, praticamente, não impõem nenhuma restrição e que permitem inclusive que o usuário faça um novo licenciamento daquele software com condições diferentes. Por exemplo, o software é livre, mas eu quero relicenciar e distribuir ele com termos não livres. Tudo bem, a licença me permite isso.*

*Nelson Lago - Especialista do IME/USP  
[http://www.tv.serpro.gov.br/Entrevista\\_Nelson\\_Lago.org/view](http://www.tv.serpro.gov.br/Entrevista_Nelson_Lago.org/view)*

#### **3.1.1. Licença BSD / Berkeley Software Distribution**

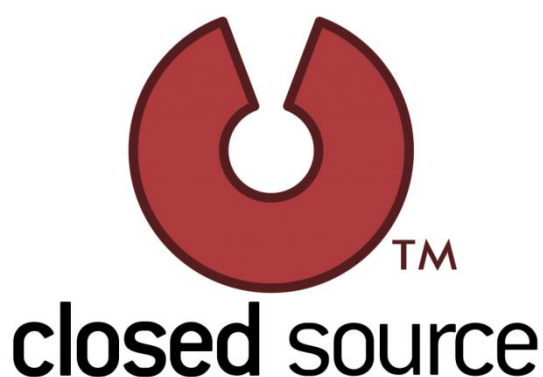
([www.opensource.org/license/bsd-license.php](http://www.opensource.org/license/bsd-license.php))

*[...] É uma licença simples, permissiva, não exige copyleft, não tem nenhum problema conhecido e é compatível com a GNU GPL. (Fonte: [www.gnu.org](http://www.gnu.org)).*

A Licença BSD se caracteriza por ter sido a primeira licença OSS escrita e, até hoje, é uma das mais usadas. Criada originalmente pela Universidade da Califórnia, em Berkeley, para seu sistema derivado do UNIX, chamado *Berkeley Software Distribution*, é atualmente usada como modelo por uma ampla gama de softwares licenciados de modo permissivo. Os principais motivos que levaram a licença BSD a ser tão difundida são a simplicidade de seu texto e o fato dela ter sido inicialmente adotada por um projeto amplamente disseminado, o que criou um círculo virtuoso e, com isso, mais comunidades a adotaram, tornando-a ainda mais conhecida.

O texto da licença é considerado como de domínio público e pode ser modificado sem nenhuma restrição.

A licença BSD permite que o software distribuído sob a licença, seja incorporado a produtos proprietários. Alguns exemplos notáveis são: o uso de código BSD em produtos da Microsoft, e o uso de muitos componentes do FreeBSD ® - um avançado sistema operacional livre do tipo Unix para servidores e estações de trabalho - no sistema Mac OS X<sup>5</sup> da Apple Computer.



**Figura 4 - Closed Source TM**

Na sua versão original, a licença BSD continha termos que a tornava incompatível com a licença GPL. Como elas estão entre as licenças mais utilizadas no mundo do software livre, a impossibilidade em combinar os seus componentes tornou-se um grande problema para os autores destes softwares. Na revisão sofrida em 1999 a cláusula controversa foi retirada. Desde esta data, os autores estão livres para incorporar softwares BSD naqueles licenciados pela GPL.

### **Vantagens e Desvantagens**

Vários autores são unânimes em afirmar que uma das principais desvantagens desta licença é a quantidade de variantes existentes, pois não fica imediatamente claro para o usuário sob quais termos um software está sendo licenciado quando a única informação que ele recebe é que está sob a licença BSD, por esse motivo, é sempre necessário verificar com cuidado sob qual versão da licença o software em questão está licenciado. Outra desvantagem, apontada por eles, diz respeito ao ponto de vista jurídico, isso ocorre porque seus termos são um tanto vagos, havendo um esforço maior de interpretação da licença para demonstrar que, de fato, a maioria dos direitos, em geral restritos ao autor, está sendo transferido para o licenciado.

Defensores da licença BSD argumentam, porém, que ela é mais livre que a licença GPL, isso se deve, segundo eles, porque ela garante quase a mesma liberdade que o domínio público e pode ser modificada sem nenhuma restrição. A licença BSD requer

---

<sup>5</sup> Mac OS X é marca registrada de Apple Inc.

apenas o reconhecimento dos autores e outras pequenas restrições. Como resultado os códigos BSD podem ser utilizados em projetos livres com outras licenças como Linux (GPL) ou softwares proprietários como Windows<sup>6</sup> e o Mac OS X. O *PostgreSQL* - Sistema Gerenciador de Banco de Dados open-source que foi iniciado em 1986, também na Universidade de Berkeley, na Califórnia - utiliza, claro, uma licença BSD.

### 3.1.2. Licença MIT/X11

([www.opensource.org/license/mit-license.php](http://www.opensource.org/license/mit-license.php))

[...] *Ela é uma licença simples, permissiva, que não exige copyleft, e é compatível com a GNU GPL.*

(Fonte: [www.gnu.org](http://www.gnu.org)).

A Licença MIT é considerada similar à Licença BSD. Conhecida em algumas outras variantes, como Licenças X11 ou Licença X, foi criada pelo Massachusetts Institute of Technology e redigida inicialmente para o X Windows System<sup>7</sup>, desenvolvido no MIT em 1987. Seu texto é bem mais explícito, do que o da licença BSD, ao tratar dos direitos que estão sendo transferidos, afirmando que qualquer pessoa que obtém uma cópia do software, e de seus arquivos de documentação associados, pode lidar com eles sem restrição, incluindo os direitos a usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do software. As condições impostas para tanto são apenas manter o aviso de *Copyright* e uma cópia da licença em todas as cópias ou porções substanciais do software.

Segundo a Wikipedia, muito grupos utilizam a licença MIT no seu software, como os programas Expat, MetaKit, PuTTY, as bibliotecas do projeto Mono, Lua, e o sistema X Window, sendo este último o exemplo mais conhecido.

#### Vantagens e Desvantagens

Um das vantagens da Licença MIT é a clareza de seus termos ao declarar, explicitamente, o que é permitido; por exemplo: sublicenciar ou vender cópias do software, que são direitos apenas implícitos na BSD.

Essa licença também é recomendada pela Free Software Foundation, para aqueles que buscam uma licença permissiva, pois é bastante conhecida e, ao contrário da BSD, não possui múltiplas versões, com cláusulas que podem gerar dificuldades adicionais.

---

<sup>6</sup> Windows é marca registrada de Microsoft Corporation.

<sup>7</sup> **X Window System**, **X-Window**, **X11** ou simplesmente **X** é um software de sistema e um protocolo que fornece uma base para interfaces gráficas de usuário (com o conceito de janelas) e funcionalidade rica de dispositivos de entrada para redes de computadores. Fonte: [http://pt.wikipedia.org/wiki/X\\_Window\\_System](http://pt.wikipedia.org/wiki/X_Window_System)

### 3.1.3. Licença Apache

(<http://www.apache.org/licenses/LICENSE-2.0>)

[...] *Esta é uma licença permissiva que não exige copyleft e que possui algumas exigências que a tornam incompatível com a GNU GPL.* (Fonte: [www.gnu.org](http://www.gnu.org)).

Criada em 2004, a licença Apache 2.0 é uma evolução de versões anteriores lançadas entre 1995 e 2000, e uma das suas principais vantagens em relação a essas versões anteriores é poder ser facilmente aplicada, sem modificações no seu texto, a qualquer projeto que deseje fazer uso de suas políticas de licenciamento – mesmo que não seja e nem pretenda ser um projeto integrante da Fundação Apache.

#### Vantagens e Desvantagens

As condições gerais da licença Apache 2.0 permitem livre uso, redistribuição e alteração, sem exigir reciprocidade – ou seja, o código pode ser reaproveitado em projetos proprietários, se esse for o interesse de algum desenvolvedor.

Ela tem algumas características que a distinguem da simplicidade espartana das BSDs, a começar pela questão das patentes de software, que fica explícita na sua terceira cláusula, esclarecendo (de forma bastante extensa e específica) que todo contribuidor de código para o software em questão concede também uma licença mundial e perpétua para uso de suas patentes que sejam necessárias para uso ou distribuição do código contribuído por ele em combinação com o software em questão.

A principal vantagem da licença Apache é seus termos estarem definidos de forma mais precisa, deixando menos margem a interpretações conflitantes com os interesses dos envolvidos.

O fato de a licença Apache deixar explícito, na cláusula sobre redistribuição, que é permitido o uso de outra licença, é visto como uma vantagem sobre as licenças BSD ou MIT quanto à clareza de sua característica permissiva.

O principal problema da Apache é o fato de sua incompatibilidade com a GPL. Vale mencionar que, apesar de toda a cooperação entre a Fundação Apache e a FSF, essa compatibilidade é unidirecional e só funciona a favor da GPL: ela permite que códigos sob a licença Apache sejam usados em projetos sob a GPL, mas impede que códigos GPL sejam usados em projetos sob a licença Apache.

Os softwares mantidos pela Fundação Apache, incluindo seu popular servidor web, vários componentes do ecossistema Java e até o *SpamAssassin*<sup>8</sup> são todos livres e de código aberto, licenciados sob a Apache License 2.0.

---

<sup>8</sup> **SpamAssassin** é um programa de computador licenciado através da licença Apache e utilizado como filtro de *spam* enviado através de mensagem eletrônica.



### 3.1.4. Outras Licenças Permissivas

(Fonte: [www.gnu.org](http://www.gnu.org))

**Licença do PHP, Versão 2.02:** Esta é uma licença de software livre que não exige copyleft e que tem problemas de ordem prática semelhantes aos da licença original do BSD, incluindo incompatibilidade com a GNU GPL.

**Licença do Python 2.0.1, 2.1.1 e versões posteriores:** É uma licença de software livre, compatível com a GPL. Entretanto, deve-se observar que versões intermediárias do Python (1.6b1 até 2.0 e 2.1) estão sob uma licença diferente.

## 3.2. Licenças Recíprocas

As Licenças Recíprocas dividem-se em Licenças Recíprocas Parciais e Licenças Recíprocas Totais. A principal diferença é que as licenças recíprocas totais se caracterizam pela manutenção dos termos da licença original (mecanismo conhecido como Copyleft). Nas recíprocas parciais essa manutenção não está assegurada.

### 3.2.1. Licenças Recíprocas Totais

As licenças recíprocas totais tem a característica de que qualquer trabalho derivado do original deve ser redistribuído e disponibilizado sob os mesmos termos da licença original. Essa filosofia é conhecida como *Copyleft* (mecanismo que veremos em detalhe mais adiante). Basicamente, a ideia do *Copyleft* é permitir a todos a execução, cópia, modificação e redistribuição das versões derivadas do software, impedindo que possam ser acrescentadas restrições a essas modificações. Essa restrição determina que todo software livre produzido pela comunidade permaneça livre e não possa ter seu código fechado. A principal licença desta categoria é a: GNU GPL que veremos a seguir.

#### 3.2.1.1. Licença GNU GPL (General Public License)

(<http://www.gnu.org/licenses/licenses.pt-br.html#TOCGPL>).

[...] *Esta é uma licença de software livre, e uma licença com copyleft. Nós a recomendamos para a maioria dos pacotes de software.* (Fonte: [www.gnu.org](http://www.gnu.org)).

É a principal licença de software livre recíproca total da Free Software Foundation. Teve sua primeira versão redigida em 1989. A versão 2.0 foi redigida

em 1991 e permaneceu até 2007 quando foi lançada a GPL v3, em vigência até os dias atuais. É bastante conhecida, sendo a licença mais utilizada em projetos de software livre no mundo, em grande parte devido à sua adoção pelo projeto GNU e o sistema operacional GNU/Linux.

A GNU GPL destina-se a garantir a liberdade de compartilhar e modificar software livre de modo a garantir que o software permaneça livre para todos os seus usuários.

Eles lembram que a GPL é desenhada para se certificar de que está assegurada a liberdade de distribuir cópias de software livre (e cobrar por elas se assim se desejar), de receber o código-fonte ou de poder obtê-lo, de poder alterar o software ou utilizar partes dele em novos programas livres, e que se saiba que se pode fazer isso.

Para proteger esses direitos, a *Free Software Foundation* precisa fazer restrições que impeçam a qualquer um negá-los ou solicitar que se abduque deles. Estas restrições traduzem-se em certas responsabilidades para quem distribui cópias do software ou modifica-o, mas a maior delas é, sem dúvida, a “responsabilidade de respeitar a liberdade dos outros”.

*[...] se você distribuir cópias de um programa, gratuitamente ou por uma taxa, você deve fornecer aos recebedores todos os direitos que você tem. Você deve se certificar de que eles também recebam ou possam obter o código-fonte. E você deve mostrar-lhes estes termos para que eles conheçam seus direitos. (Fonte: [www.gnu.org](http://www.gnu.org)).*

A GNU-GPL protege os direitos dos autores em duas etapas:

1ª Com copyright do software;

2ª Oferecendo-lhes uma licença que lhes dá permissão legal para copiar, distribuir, bem como, modificar o software.

Vamos entender como isso funciona:

Para a proteção de cada autor, a GPL claramente explica que não há nenhuma garantia para o software livre e, para a segurança tanto do autor quanto dos usuários, ela requer que versões modificadas sejam marcadas como tal, de modo que, nenhum possível problema seja atribuído erroneamente aos autores das versões anteriores.

*[...] Se o software é modificado por alguém e passado adiante, é fundamental que seus destinatários saibam que o que eles têm em mãos não é o software original, de modo que quaisquer problemas introduzidos por terceiros não afetem as reputações dos autores originais. (Fonte: [www.gnu.org](http://www.gnu.org)).*

Ciente de que qualquer programa livre é constantemente ameaçado por patentes de software, a GPL assegura que patentes não podem ser usadas para tornar um programa livre em um programa não livre.

[...] *Queremos evitar o perigo de que distribuidores de software livre obtenham patentes individuais, tornando o programa proprietário. Para evitar isso, temos claro que qualquer patente deve ser licenciada para uso livre de todos.*  
(Fonte: [www.gnu.org](http://www.gnu.org)).

A FSF orienta quem pensa em desenvolver um novo programa, e deseja que ele seja da maior utilização possível para o público, que a melhor maneira de alcançar este objetivo é torná-lo um software livre que qualquer um pode redistribuir e alterar, sob os termos da GNU GPL que podem ser encontrados no site da GNU General Public License.

### **Vantagens e Desvantagens**

Segundo a Wikipedia, os defensores do GPL argumentam que a obrigatoriedade de que softwares derivados sejam licenciados sobre GPL fomenta o crescimento do software livre. Apesar de a licença BSD ser evidentemente mais "livre" que a GPL, pode-se argumentar que a GPL é mais "livre" no sentido em que garante a liberdade nos trabalhos derivados. Por causa desse conceito de "liberdade" do copyleft, pode-se dizer então que a BSD é mais "liberal" que a GPL. De certa forma, pode-se considerar que:

[...] *a GPL é "interesseira", mas seria um interesse "benéfico", no sentido de que é interessante para a comunidade GPL manter os trabalhos derivados sob a GPL também.*  
(Fonte: [http://pt.wikipedia.org/wiki/Licen%C3%A7as\\_BSD\\_e\\_GPL](http://pt.wikipedia.org/wiki/Licen%C3%A7as_BSD_e_GPL)).

### **3.2.1.2. Licença AGPL (GNU Affero General Public License)**

É uma licença de software livre publicada recentemente pela Free Software Foundation, foi desenvolvida pela Affero Inc. como uma derivação autorizada da licença GPL com o intuito de permitir o uso de um software através de uma rede.

A Affero Inc. ([www.affero.org](http://www.affero.org)) é uma empresa que se define com a missão de “trazer a cultura de patrocínio para a Internet”. Para a AGPL valem todas as considerações feitas a respeito da GPL. Ela é recomendada para projetos em que há interação via rede e busca-se o copyleft.

A AGPL tem o propósito de ser uma licença minimamente modificada da GNU GPL e atender as necessidades de fornecer liberdade em softwares como serviços (SaaS, Software as a Service), ou seja, aqueles os quais você não tem acesso direto ao binário/código-objeto.

A GNU Affero GPL requer que o código-fonte deva estar completamente disponível para qualquer usuário do software coberto pela GNU AGPL, tipicamente uma aplicação web.

A Free Software Foundation recomenda essa licença para qualquer software o qual irá ser rodado atrás de um servidor de mesmo modo como funciona uma aplicação web.

### **Vantagens e Desvantagens**

Para a AGPL valem todas as considerações feitas a respeito da GPL. Ela é recomendada para projetos em que há interação via rede e busca-se o Copyleft. A AGPL é considerada a mais Viral das Licenças – como são chamadas as Licenças Copyleft, porque quaisquer trabalhos derivados de outro sob a licença Copyleft são, por obrigatoriedade, regidos pela mesma quando distribuídos (exibindo assim um comportamento viral) - portanto deve ser evitada em projetos em que haja qualquer expectativa de utilização sob outra licença, a não ser que seja adotado um modelo de múltiplo licenciamento.

### **3.2.2. Licenças Recíprocas Parciais**

As licenças recíprocas parciais também são conhecidas como Licenças *Copyleft Fraco* (termo que também desenvolveremos mais adiante). Em princípio determinam que modificações feitas em um software sob esta licença sejam disponibilizadas sob a mesma licença. Entretanto, se as modificações forem utilizadas como componentes de outro projeto de software, este projeto não precisa, necessariamente, ser disponibilizado sob a mesma licença. Este é o principal diferencial das licenças recíprocas parciais para as recíprocas totais.

Essas licenças possuem um equilíbrio entre as licenças permissivas e as recíprocas totais. Considera-se que as licenças recíprocas parciais proporcionam o interesse da comunidade de desenvolvedores pela liberdade que promovem. São exemplos destas categorias as licenças:

### **3.2.2.1. Licença LGPL (GNU Lesser General Public License)**

([www.fsf.org/licensing/licenses/lgpl.html](http://www.fsf.org/licensing/licenses/lgpl.html))

*[...] Esta é uma licença de software livre, mas não é uma licença com copyleft forte, pois ela permite a ligação com módulos não livres. Ela é compatível com a GNU GPL. Mas nós a recomendamos somente em circunstâncias especiais. (Fonte: [www.gnu.org](http://www.gnu.org)).*

Originalmente redigida como uma cópia da licença GPL com alterações relativas a bibliotecas, definidas como um conjunto de funções de software que podem ser agregadas para formar novos softwares, a LGPL foi reescrita em 2007 para ficar em conformidade com a GPLv3.

O Projeto GNU tem duas licenças principais a serem usados para as bibliotecas. Um deles é o GNU Lesser GPL (LGPL) e a outra é o GNU GPL:

- Usar a GNU LGPL permite a utilização da biblioteca em programas proprietários;
- Usar a GPL comum permite a utilização da biblioteca apenas em programas livres.

#### **Vantagens e Desvantagens**

A LGPL é entendida como uma licença bastante complexa, que requer uma observação atenta dos seus termos para evitar o descumprimento que pode acarretar em uma ação judicial, mas apesar da dificuldade em usar a LGPL, ela é uma licença amplamente adotada, pois combina características permissivas e recíprocas de forma balanceada, trazendo as vantagens de ambos os modelos.

### **3.2.2.2. Licença Mozilla/MPL (Mozilla Public License)**

(Fonte: <http://www.mozilla.org/foundation/>)

A licença Mozilla é considerada bem escrita e serviu como modelo para muitas das licenças de software livre comercial que a seguiram. Ela une características de licenças recíprocas e de licenças permissivas e, portanto, também é categorizada como uma licença recíproca parcial. Ela determina que o código sob esta licença deva ser redistribuído pelos termos da mesma. Entretanto, o código pode ser utilizado em projetos maiores, que podem estar sob outra licença.

#### **Vantagens e Desvantagens**

Devido à clareza de seus termos, baseados em definições precisas, a MPL é uma licença mais fácil de entender e aplicar do que a LGPL. Portanto, quando se busca

uma licença que combine as vantagens do modelo recíproco com o modelo permissivo, a MPL é uma ótima alternativa.

Porém, uma desvantagem da MPL em relação à LGPL é a incompatibilidade com a GPL. Não é possível juntar projetos que estejam sob as licenças MPL e GPL, pois a MPL obriga que o código original mantenha-se como MPL e a GPL obriga que o trabalho como um todo e, conseqüentemente, cada uma de suas partes seja GPL.

### **3.2.2.3. Licença EPL (Eclipse Public license)**

(Fonte: <http://www.eclipse.org/>)

*[...] O Eclipse Public License (EPL) é uma licença de software de código aberto usado pela Fundação Eclipse para o seu software. Ele substitui a Licença Pública Comum (CPL) e remove certos termos relativos a litígios relacionados a patentes.*

(Fonte: [http://en.wikipedia.org/wiki/Eclipse\\_Public\\_License](http://en.wikipedia.org/wiki/Eclipse_Public_License)).

A EPL foi criada pela Eclipse Foundation para substituir duas licenças anteriores à criação da fundação: a IBM Public Licence de 1999 que foi posteriormente substituída pela Common Public Licence.

Existem dois tipos de atores na EPL. Eles são os chamados "colaboradores" e "destinatários". Entre os colaboradores estão o colaborador inicial (desenvolvedor), que é a pessoa ou entidade que cria o código fonte que será distribuído sob a EPL; há também os colaboradores subsequentes, que são todos aqueles que efetuam alterações ou fazem adições ao código inicial. Qualquer pessoa ou entidade que redistribui o programa é também chamada colaborador. Os destinatários são todos aqueles que recebem o programa sob a Licença EPL, incluindo os colaboradores.

Os colaboradores de programas licenciados sob a Licença EPL entendem que o código fonte dos programas será disponibilizado nos termos da EPL. A menos que você seja o proprietário do software ou tenha recebido permissão do proprietário, você não está autorizado a aplicar os termos de uma licença adicional para o programa, incluindo-o, por exemplo, em um programa licenciado sob outra licença Open Source.

A Licença EPL está devidamente aprovada pela OSI e é listada como uma licença de Software Livre pela FSF.

### 3.3. Comparativo das principais licenças de Software Livre e/ou OSS

SABINO, V. C., em Um estudo sistemático de licenças de software livre, 2011. 116f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2011 – apresenta um quadro comparativo das principais licenças de Software Livre e/ou Open Source.

	BSD	MIT	Apache	GPL 2.0	GPLV3	AGPL	LGPL	Mozilla / EPL
Reciprocidade	Ausente	Ausente	Ausente	Total	Total	Total	Parcial	Parcial
Clareza / Simplicidade	Média	Alta	Alta	Média	Média	Média	Baixa	Alta
Adoção	Alta	Baixa	Média	Alta	Média	Baixa	Alta	Alta
Compatibilidade com a GPL	Sim	Sim	V3	2.0	V3	Sim	Respectiva	Não
Licenças derivadas	Sim	Sim	Sim	Não	Não	Não	Não	Sim

Tabela 1 – Quadro Comparativo das Principais Licenças de Software Livre e OSS

**Reciprocidade:** Pode ser de três tipos, conforme abaixo:

- **Ausente:** não exige licenciamento de trabalhos derivados sob os mesmos termos;
- **Parcial:** dispensa o licenciamento equivalente se o projeto original for componente do novo projeto;
- **Total:** obriga qualquer trabalho derivado a ser licenciado sob os mesmos termos.

**Clareza / Simplicidade:** Destaca que é necessária atenção especial aos termos omissos e interpretações diversas em licenças de assertividade média ou baixa.

**Adoção:** Licenças com maior grau de adoção facilitam a disseminação do projeto entre usuários e empresas e novos desenvolvedores, na medida em que aumentam a chance de contato prévio com as mesmas.

**Compatibilidade com a GPL:** A compatibilidade com essa licença em particular é um critério importante, pois ela é uma das mais usadas em projetos já existentes, e, ao mesmo tempo, uma das que mais apresenta restrições de reciprocidade.

**Licenças Derivadas:** Licenças que permitem a criação de derivadas podem oferecer maior liberdade, mas tornam a verificação de compatibilidade menos trivial.

### 3.4. Metodologia para Escolha de Licenças em um Projeto de Software Livre

Sabino (2011, pag.56) apresenta ainda em seu trabalho uma metodologia para escolha de licenças em um novo projeto de Software Livre. A autora resalta que essa metodologia é apenas uma sistematização das principais características das licenças e que não abrange detalhes de implementações específicas dos softwares.

A Figura 5 representa a metodologia proposta na forma de um fluxograma.

Segundo Sabino (2011, pag. 55) o processo é bastante simples e inicia na decisão de distribuir um programa como software livre (1).

- No caso de se desejar uma contribuição a uma comunidade já existente (2), por exemplo, um *plugin* para o ambiente Eclipse, o ideal é utilizar a mesma licença do projeto principal dessa comunidade, de forma a evitar qualquer incompatibilidade atual ou futura. Porém, a autora destaca que, se for um projeto mais independente, não atrelado de nenhuma forma a alguma licença preexistente, o principal fator de decisão é até que ponto o detentor dos direitos quer impor limitações no seu uso e distribuição.
- Se a ideia é que o projeto seja utilizado apenas em software livre (3), fortalecendo a comunidade, sem se importar com a incompatibilidade com projetos que utilizem outras licenças, então, deve ser utilizada a GPL. A escolha da versão também depende dos propósitos e dos componentes utilizados e, no caso do projeto querer se beneficiar de maior adoção, o ideal é escolher uma licença com menor dificuldade de compatibilização.
- A autora ainda afirma que se for importante que alterações do projeto sejam atribuídas aos autores, mas assegurando que futuras versões do seu software também sejam livres (4), então, a licença a ser escolhida deve ser uma recíproca parcial, que pode ser a LGPL, para facilitar a compatibilidade com a GPL, ou alguma das outras licenças que seguem esse modelo, como por exemplo, a Mozilla.
- Porém, caso deseje assegurar a compatibilidade com a GPL 2.0, então deve-se escolher a MIT ou a nova licença BSD. Estas licenças normalmente são usadas quando se deseja que o projeto possa atingir um número maior de pessoas.
- Negadas as premissas anteriores, a Licença Apache v2 ou qualquer outra licença permissiva, é a opção.



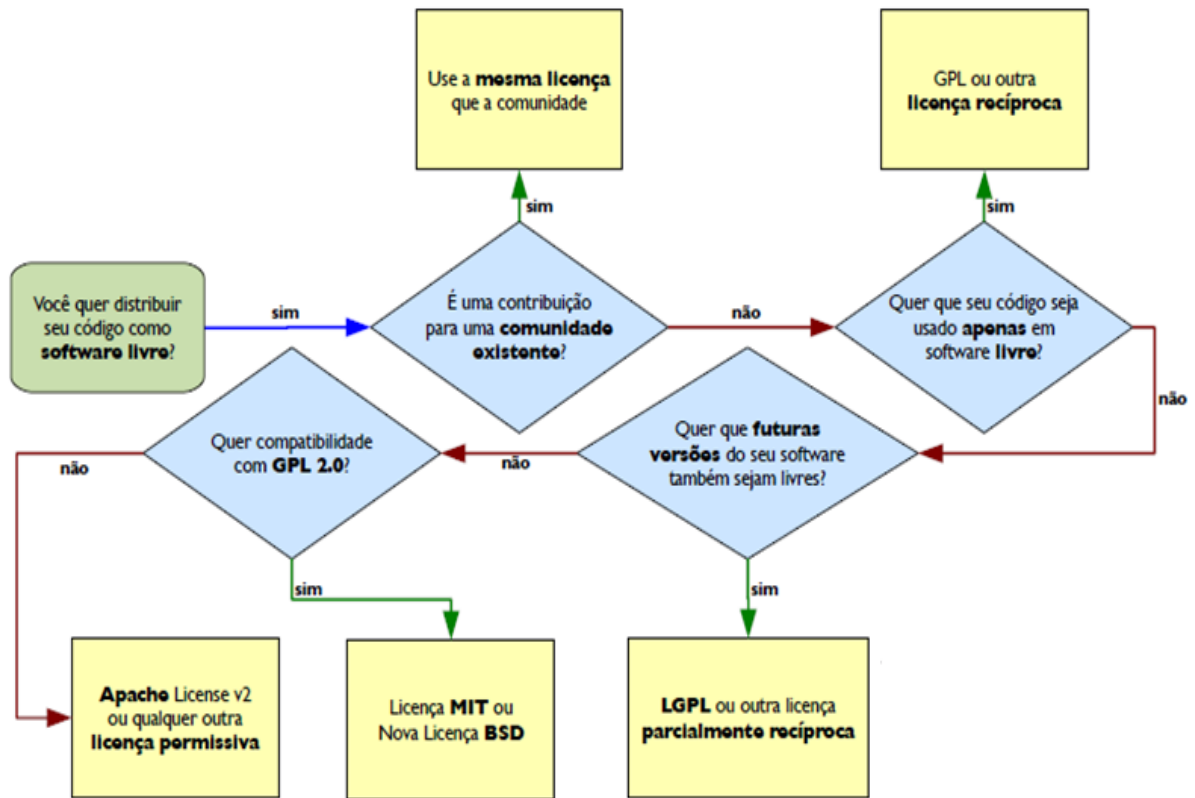


Figura 5 - Metodologia para escolha de licenças

### 3.5. Comparativo entre as principais licenças de software livre e de código aberto

Apesar das diferenças filosóficas fundamentais entre o movimento do Software Livre e o movimento Open Source, as definições oficiais de software livre pela FSF e do software de código aberto pela OSI referem-se, basicamente, as mesmas licenças de software, com poucas exceções. Embora sublinhando as diferenças filosóficas, os comentários da FSF:

[...] *O termo OSS é usado por algumas pessoas para significar mais ou menos a mesma categoria do termo Software Livre. Não é exatamente a mesma classe de software: A FSF aceita algumas licenças que nós consideramos muito restritivas, e existem licenças de Software Livre que eles não aceitam. Todavia, as diferenças na extensão da categoria são pequenas: quase todo Software Livre é Open Source e quase todo Software Open Source é Livre.* (Fonte: [www.fsf.org](http://www.fsf.org)).

Para Taurion (2004, p.23), o OSD não é simplesmente uma modalidade de licenciamento como o próprio autor descreve:

[...] *não é uma modalidade de licenciamento, mas um conjunto de especificações, regras e diretrizes para licenças baseadas no modelo de software livre. É diferente do modelo GPL, pois não obriga que o código alterado seja disponibilizado publicamente. As licenças OSD não sofrem o efeito de contaminação. O OSD permite liberdade às iniciativas que agrupem software livre e proprietário na mesma solução. É uma visão pragmática do mundo real dos negócios, onde o software livre se insere perfeitamente.*

Elias (2007, pag. 68) apresenta uma tabela comparativa entre as principais licenças de software livre e de código aberto. As licenças BSD e MPL são as mais afetadas pelas diretrizes da OSD. A licença LGPL se diferencia das demais apenas por não permitir que modificações efetuadas nos códigos, por outrem, não possam retornar como contribuição para o projeto original.

LICENÇAS	Pode ser agregado com software Não livre	As modificações efetuadas por outros podem não retornar ao software original	Pode ser re-licenciado por outros
GPL			
LGPL	X		
BSD	X	X	
MPL/Mozilla	X	X	
Domínio Público	X	X	X

Tabela 2 - Tabela comparativa das práticas de licenciamento entre software livre e código aberto

## 4. Obrigações e Riscos Envolvidos

A BARDEHLE Pagenberg, uma das maiores empresas de advocacia da Europa, especializada em “Propriedade Intelectual” (IP – *Intellectual Property*) destaca, em seu site, que a IP está entre os ativos mais valiosos de uma organização. E afirma:

[...] *Uma vez que a empresa garantiu a proteção jurídica de seus "ativos intangíveis", que pode explorá-los economicamente, por exemplo, por meio de licenciamento estratégico, isso pode se tornar um fator decisivo para seu sucesso econômico. Embora hoje existam mais negócios de licenciamento IP do que nunca, essas operações se tornam cada vez mais globais e complexas, e as tecnologias subjacentes também estão cada vez mais complexas. Portanto, o licenciamento de IP deve ser implementado com muito cuidado e de forma bastante profissional.*

As diversas publicações, livros e artigos profissionais, disponibilizados ou apresentados pela BARDEHLE Pagenberg em sua página na Internet no endereço: <http://www.bardehle.com/en/home.html>, foram essenciais para a construção deste capítulo.

## 4.1. Entendendo o Mecanismo de “Copyleft” Open Source License

(Fonte: [www.gnu.org/copyleft](http://www.gnu.org/copyleft))

Softwares contam com a proteção Copyright em praticamente todos os lugares do mundo. Normalmente o copyright é usado por um autor para proibir outros de usar, copiar, modificar ou distribuir o trabalho do autor sem a sua autorização. Além disso, para os softwares proprietário, muitas vezes apenas o código objeto do software é disponibilizado para os usuários, o que dificilmente oferece possibilidades de adaptação. Em contraste, uma licença OSS prevê que todos os que recebem uma cópia do software tem permissão para usar livremente, copiá-lo ou modificá-lo. Para permitir que o destinatário do OSS possa adaptá-lo para seus propósitos e desenvolvê-lo ainda mais, os autores do OSS fornecem o código fonte dos programas.

Com relação às regras que se aplicam a redistribuição de OSS, e em particular das regras aplicáveis à distribuição de programas derivados de OSS (ou seja, modificações de OSS), existem basicamente duas categorias de licenças:

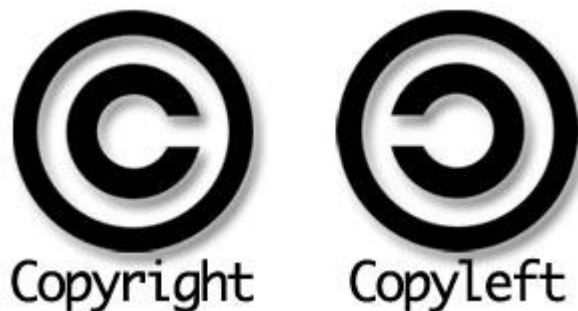


Figura 6 - Copyright e Copyleft

A primeira categoria de licenças de OSS segue o chamado paradigma "copyleft". Uma licença Copyleft normalmente prevê que o destinatário de OSS pode redistribuir o OSS, ou programas derivados do OSS, apenas sob a condição de que:

- (1) ele também fornecerá o código fonte do programa;
- (2) a redistribuição será feita sob uma licença Open Source, não sob uma licença proprietária.

O conceito de Copyleft usa a lei de copyright para garantir que um trabalho é sempre redistribuído com seu código-fonte e, portanto, continua a ser adaptável. Enquanto o OSS pode ser vendido, os destinatários da OSS são livres para redistribuí-lo, sem pagar royalties para o autor original. Como vimos anteriormente, a GNU General Public License (GPL) é um exemplo típico de uma licença Copyleft.

A segunda categoria de licenças de OSS, as chamadas licenças Não-Copyleft, têm exigências menos rigorosas para a redistribuição da OSS ou distribuição de programas derivados de OSS. Enquanto estas licenças podem impor certas obrigações menores que devem ser observadas para a (re) distribuição, elas não exigem que o código fonte seja disponibilizado e permite que as obras derivadas sejam distribuídas sob uma licença proprietária. Exemplos típicos de licenças Não-Copyleft OSS incluem a licença Apache e as licenças BSD.

Muitas licenças de OSS podem ser colocadas em algum lugar entre um Copyleft puro e uma licença Não-Copyleft. Por exemplo, o GNU Lesser General Public License (LGPL) permite que qualquer tipo de ligação com software proprietário, ou qualquer tipo de utilização, no caso o software sob a LGPL é uma biblioteca, mas não permite que um programa que modifica o código do software original seja distribuído sob uma licença proprietária.

Como as licenças Copyleft envolvem o maior risco para as empresas quando um OSS é usado por elas ou integrado a programas proprietários das mesmas, este capítulo irá se concentrar em licenças Copyleft puras. O GPL é de longe a mais utilizada licença OSS Copyleft, e é por isso que a GPL foi utilizada como base para análise.

## **4.2. Obrigações impostas ao licenciado de OSS.**

Antes de qualquer coisa, é importante ter claro o entendimento de que uma licença Open Source somente impõe obrigações ao licenciado quando o OSS é distribuído para terceiros, mas não quando ele é usado ou modificado para fins próprios do titular da licença.

A licença GPLv3 fala, especificamente, em "transmitir" e, de acordo com a definição estabelecida por ela, "transmitir" um trabalho significa qualquer tipo de "propagação" que permita a outros criar ou receber cópias. Este termo abrange receber, realmente, o software em um meio tangível (por exemplo, um CD) e baixar o software. A mera interação com o usuário através de uma rede de computadores, sem a transferência de uma cópia, não é considerado transmissão.

Basicamente, temos um conjunto de obrigações impostas às organizações quando uma licença copyleft OSS, ou trabalhos derivados de licenças copyleft OSS, são distribuídos a

terceiros. Estão listadas abaixo as mais importantes e cujo conhecimento é essencial para todos aqueles que utilizam e/ou redistribuem OSS:

1. **Fornecer o código fonte:** Programas OSS ou programas derivados de OSS, distribuídos pelas organizações a terceiros podem ser transmitidos na forma de código-fonte ou sob a forma de código objeto, mas neste caso, os códigos-fonte devem, mandatoriamente, ser fornecidos.  
O código de fonte deve ser sempre a forma na qual um programador iria alterar o programa. Não é permitido ofuscar, deliberadamente, o código fonte. No entanto, o código fonte não inclui as bibliotecas do sistema, geralmente disponíveis gratuitamente e que são utilizadas, sem modificações, para realizar as atividades.
2. **Permitir modificação sem custo, bem como, sua redistribuição:** As empresas podem vender/cobrar pelos OSS e trabalhos derivados do OSS para seus clientes. No entanto, o software deve ser livremente redistribuível. Isto significa que a licença dada aos clientes, deve permitir-lhes redistribuir livremente o software fornecido sob uma licença Open Source, bem como seus próprios trabalhos derivados a partir deste software. As empresas não podem cobrar royalties ou outra taxa para tais redistribuições de seus clientes
3. **Fornecer uma cópia dos termos de licença Open Source:** Um distribuidor de OSS deve, visivelmente e de forma adequada, publicar, em cada cópia, uma nota de copyright apropriada e fornecer aos destinatários uma cópia dos termos de licença Open Source, juntamente com cada programa OSS. Além disso, se OSS é modificado antes de ser redistribuído, os arquivos modificados devem conter avisos proeminentes afirmando que eles são modificados, o nome da parte que alterou os arquivos, bem como a data relevante da mudança.

#### **4.3. Riscos envolvidos com o descumprimento dos Termos e Condições de Licenciamento OSS**

Ainda segundo a BARDEHLE Pagenberg, é preciso considerar fortemente os riscos que surgem para as organizações quando as mesmas descumprem os termos e condições de uma licença OSS. Entender o que ocorre, por exemplo, quando uma empresa distribui um programa OSS ou programas derivados de OSS para seus clientes sem fornecer o código fonte e/ou sob uma licença proprietária em vez de fornecê-los sob a licença OSS.

É importante analisar as consequências jurídicas de uma violação dos termos de uma licença OSS, ou seja, se tal violação é considerada como uma violação das obrigações contratuais ou como uma violação de direitos autorais.

1. **Consequência jurídica da violação dos termos de uma licença OSS:** Em princípio, duas diferentes consequências jurídicas de uma violação de termos de licenciamento são possíveis, ou seja, a não conformidade com os termos da licença poderá ser tratado como:
  - (1) Violação de Contrato ou;
  - (2) Violação de Direitos Autorais.

Determinar se a primeira ou a segunda abordagem ocorreu é essencial, pois obviamente demandam soluções diferentes. Apenas no caso de Violação de Direitos Autorais pode uma organização ser legalmente proibida de distribuir um OSS. Por outro lado, a possibilidade da empresa ser obrigada a abrir o código fonte de programas OSS ou derivados de OSS, distribuídos aos seus clientes, só existe sob as regras de Violação de Contrato.

### **(1) Primeira abordagem: Violação de Contrato**

A licença OSS é considerada como um contrato de licença celebrado entre o autor original do OSS (o licenciador) e o destinatário da OSS (o licenciado), que é quem redistribui o OSS ou programas derivados de OSS.

Uma licença de software livre, por exemplo, a GPL, constitui um contrato padrão, com termos e condições pré-formulados. O acordo será celebrado quando licenciado fizer o download do OSS ou quando ele mandar uma cópia do software, desde que ele seja simultaneamente informado sobre as condições impostas pela licença OSS.

Não conformidade com as regras estabelecidas acima poderão ser tratadas sob as regras de Violação de Contrato.

### **(2) Segunda abordagem: Violação de Direitos Autorais**

Do mesmo modo, uma não conformidade com os termos da licença OSS, por parte do licenciado, pode resultar em uma Violação dos Direitos Autorais do autor original do software. Duas teorias podem levar à adoção desta segunda abordagem:

- **Primeira Teoria:** A licença de OSS é considerada como um contrato entre o autor original e o licenciado, o qual está sujeito à condição de usuário pelos termos de licença. No caso de descumprimento, o contrato de licença é automaticamente rescindido.

A licença de software livre também pode ser considerada como uma concessão unilateral de direitos, sob a condição de que o usuário aceita os termos de licença. No caso de descumprimento, o subsídio é automaticamente revogado. Quando o contrato de licença terminar ou quando a concessão unilateral de direitos é revogada, todos os direitos anteriormente concedidos ao licenciado voltam para o autor original. Neste caso, o licenciado não teria mais o direito de executar, copiar, modificar ou distribuir o OSS. Consequentemente, a distribuição do OSS para os seus clientes infringiria direitos autorais exclusivos do autor original.

Como não há obrigações contratuais restantes, fica caracterizada a Violação de Direitos Autorais, mas não a Violação de Contrato.

- **Segunda teoria:** Um redistribuidor de OSS excede o âmbito da licença (em especial o direito de redistribuir) quando ele redistribui sem fornecer o código-fonte ou se redistribuir sob uma licença proprietária. O direito de redistribuir está condicionado ao fornecimento do código fonte sob a licença Open Source. Portanto, em casos de descumprimento, entende-se que o Direito Autoral do autor original é violado.
2. **Demais pontos a considerar:** Normalmente, todo aquele que contribuiu com uma parte do OSS será dono do direito autoral da sua contribuição. Mas há casos em que todos os contribuintes, em conjunto, poderão possuir os direitos autorais do OSS. Segundo a BARDEHLE Pagenberg, no direito alemão, em ambos os casos cada colaborador, individualmente, pode abrir processo contra um infrator. Ainda segundo estes, especialmente nos EUA, os direitos dos proprietários dos direitos autorais estão, frequentemente, ligados a uma organização. Por exemplo, a Free Software Foundation exige que os contribuintes para os seus projetos de FSF atribuam os direitos autorais à FSF.

Considera-se que um programador, individualmente, dificilmente abrirá um processo contra empresas por violarem licenças OSS, até porque, dificilmente ele terá os meios financeiros necessários para fazê-lo. No entanto, organizações como a FSF e a OSI, cujo objetivo é garantir que os termos de licença do GPL são observados, são reconhecidamente ativas.

Em 2008, a FSF processou a Cisco Systems, Inc. por Violação de Direitos Autorais, alegando que a empresa usava, em um de seus produtos, software do projeto GNU, sem estar em conformidade com a GPL. A FSF solicitou que a Cisco fosse proibida de usar o OSS. Segundo a Wikipédia, o caso só foi resolvido depois que a Cisco se comprometeu a cumprir todos os termos da licença GPL, além de fazer uma contribuição monetária (não divulgada) para a FSF ([http://en.wikipedia.org/wiki/Free\\_Software\\_Foundation\\_v.\\_Cisco\\_Systems](http://en.wikipedia.org/wiki/Free_Software_Foundation_v._Cisco_Systems)).

#### 4.4. Programas derivados de OSS

Está claro que as empresas tem que cumprir os termos das licenças Open Source quando elas redistribuem um original OSS não modificado. No entanto, como tem sido repetidamente mencionado, os termos de distribuição da licença Open Source também se aplicam a programas derivados do OSS original Esta seção tem como objetivo dar uma visão geral de quando um programa é classificado como uma obra derivada de OSS.

## **I. Modificação do OSS original**

No caso da organização utilizar um programa Open Source e modificar seu código, por exemplo, por correção ou melhoria do OSS, o programa resultante é qualificado como uma obra derivada e claramente cai sob a licença Open Source.

Para criar um trabalho verdadeiramente independente, as modificações teriam de ser de tão grande alcance que o código fonte modificado dificilmente pudesse ser reconhecido como sendo baseado no código original.

Nenhuma obrigação é aplicável enquanto a empresa modifica o OSS e executa os programas modificados apenas internamente. No entanto, assim como a empresa distribui a versão OSS modificada para seus os clientes, as mesmas regras se aplicam para a redistribuição de OSS não modificado. Isto significa que os programas modificados só podem ser distribuídos sob a licença Open Source, nunca sob uma licença proprietária, além disso, a organização deve fornecer o código fonte do programa modificado.

## **II. Integração do OSS com Software Proprietário**

Um OSS pode ser integrado com um software proprietário de certo número de maneiras diferentes. Quando isso for feito, o perigo para a organização é que o trabalho integrado resultante pode ser qualificado como uma obra derivada do OSS original. Neste caso, o trabalho integrado só poderá ser distribuído sob a licença Open Source, e o código-fonte do trabalho integrado deverá ser fornecido na íntegra. Este é o chamado "efeito viral" de OSS.

## **III. Integração das OSS código fonte em código fonte proprietário**

Se código OSS é integrado no código fonte proprietário, de modo copiar e colar, a obra resultante é susceptível de ser considerado como derivado tanto do código proprietário quanto do código OSS. Como resultado disso, o código proprietário está "infectado" pelo código OSS no sentido de que a obra resultante pode ser distribuída somente sob uma licença Open Source, e somente se o código fonte de todo o trabalho for fornecido. Uma exceção se aplica quando a quantidade de código copiado OSS é tão pequena (talvez menos de 20 linhas de código) que nenhuma das partes copiadas tem um nível suficiente de originalidade e, portanto, não gozam de proteção de direitos autorais.

## **IV. Os programas de OSS e proprietárias interoperam através de Interfaces**

Por outro lado, se a "integração" do OSS e do programa proprietário consiste apenas numa interface entre o programa proprietário e o OSS em tempo de execução, de forma bem definida (por exemplo, um programa aplicativo proprietário rodando em Linux), a OSS e o trabalho proprietário permanecem separados do ponto de vista de direitos autorais. Isto significa que nenhum trabalho derivado foi criado, mesmo se o OSS e o programa proprietário são entregues ao cliente em um pacote.



Neste caso, a licença Open Source não cobre o programa proprietário. No entanto, um pacote contendo tanto o OSS e o programa proprietário só poderá ser distribuído se a parte OSS do pacote está licenciada sob o OSS e se o código fonte do OSS do pacote é fornecido. O programa proprietário pode ser licenciado sob uma licença proprietária, e nenhum código-fonte do programa proprietário deve ser fornecida.

## V. Outros Casos

Há casos entre a integração do código (como descrito acima no item 01) e a separação dos programas (como descrito no item 02) que são mais difíceis de avaliar.

Existem opiniões diferentes sobre a questão de que links estáticos e/ou dinâmicos de bibliotecas OSS com softwares proprietários teriam o efeito de alocar todo o trabalho resultante sob uma licença Open Source. Acredita-se que nenhum tipo de link OSS pode infectar um software proprietário, desde que o código-fonte do OSS continue sem alteração. Para evitar o “efeito viral” do OSS, é muito importante que o código fonte do OSS e código-fonte do software proprietário permaneçam separados.

## 5. Conclusão

Segundo Aniszczyk, *Software Livre* é entendido como todo aquele software disponibilizado com uma licença de uso que busca assegurar que determinados direitos sejam concedidos aos seus usuários. O direito primordial determina que o código fonte seja sempre disponibilizado, de modo que seus usuários possam estudar, modificar e redistribuir o software se assim o desejarem. A proteção dos direitos dos usuários é obtida com um mecanismo denominado Copyleft: a licença do software reivindica a proteção Copyright e proíbe distribuição, exceto para os usuários que tenham os respectivos direitos. O Copyleft também requer que qualquer software redistribuído seja coberto pela mesma licença.

Ainda segundo esse autor, muito do medo, incerteza e dúvida que se disseminou sobre software livre envolve a, assim chamada, “natureza viral” das licenças de Copyleft — a ideia de que usar software livre como parte de um programa que você desenvolver, implica na perda da propriedade intelectual, porque a licença “infectaria” as partes proprietárias desenvolvidas. Em outras palavras, a licença pode exigir que todo software empacotado com o software livre, incluindo qualquer software recém-desenvolvido, deva ser liberado sob a mesma licença. Embora isso possa ser verdade para a maioria das licenças de Copyleft bem conhecidas, a Licença Pública Geral GNU (de acordo com a qual o Linux®, por exemplo, é liberado), há outras licenças que fornecem um equilíbrio diferente entre as preocupações comerciais e a comunidade.

O estudo desenvolvido para este artigo permite concluir que se uma organização distribui versões (modificadas ou não) de *Open Source Software* em seus produtos, para seus clientes, deve fazê-lo sob a respectiva licença OSS, que permita aos clientes redistribuir o OSS livremente. Além disso, a organização tem que fornecer o código fonte (modificado ou não) do software e uma cópia da licença *Open Source*.

Para as soluções onde o Software Open Source está integrado com o Software Proprietário da organização, ela deve manter o código-fonte do OSS e o código-fonte do Software Proprietário de forma estritamente separada. Enquanto essa regra for observada, entende-se que não há obrigação de se expor o código fonte do software proprietário.

Se uma organização distribuir a parte OSS de um trabalho integrado de acordo com as regras da licença OSS e fornecer o código fonte da parte OSS para seus clientes, é bastante provável que ele esteja em conformidade com a licença Open Source. Vale a leitura cuidadosa da licença aplicável.

Resumindo, é permitido às empresas utilizarem OSS para qualquer propósito, e também fazer alterações no software, contanto que as mesmas só utilizem os OSS internamente e não os distribuam a terceiros, não tendo, deste modo, obrigações a cumprir. No entanto, se a empresa distribui OSS ou programas derivados de OSS aos seus clientes, o código fonte do programa e os termos da licença de código aberto devem ser fornecidos com o OSS. Além disso, enquanto as empresas podem cobrar um preço para a distribuição de OSS, os clientes por sua vez devem ter permissão para copiar livremente e/ou redistribuir o software sem a obrigação de pagar royalties para as mesmas.

## Bibliografia

**ALENCAR**, Anderson Fernandes de, et al. Software livre, cultura hacker e o ecossistema da colaboração / organização Vicente Macedo de Aguiar ; ilustrações Murilo Machado. -- São Paulo: Momento Editorial, 2009. Vários autores.

**ANISZCZYK**, Chris; **GALLARD**, David. Introdução à Plataforma Eclipse. Disponível em: <http://www.ibm.com/developerworks/br/library/os-eclipse-platform/index.html?ca=drs>. Acessado em 12 de Maio de 2012.

**ARANTES**, Alison Carmo. *Comparativo de licenças de código aberto*, Belo Horizonte 2009.

**BR-Linux.org** [Internet] O que é software livre. Disponível em: <http://br-linux.org/faq-softwarelivre/>. Acessado em 10 de Maio de 2012.

**COSTA**, Marcos da; **MARCACINI**, Augusto Tavares Rosa. *Primeiras linhas sobre o software livre*, Setembro/2003.

**FALCÃO**, Joaquim et al. Estudo sobre o software livre. Escola de Direito da Fundação Getúlio Vargas. 2005.

**GNU.ORG** [Internet] Várias Licenças e Comentários sobre elas. Disponível em: <http://www.gnu.org/licenses/license-list.html>. Acessado em 12 de Maio de 2012.

**BARDEHLE.COM** [Internet] *Getting the deal through Licensing 2012* Disponível em: [http://www.bardehle.com/en/publications/search\\_in\\_all\\_publications.html](http://www.bardehle.com/en/publications/search_in_all_publications.html). Acessado em 05 de Maio de 2012.

**BARDEHLE.COM** [Internet] *Licensing*. Disponível em: [http://www.bardehle.com/en/legal\\_areas/licensing.html](http://www.bardehle.com/en/legal_areas/licensing.html) Acessado em 20 de Abril de 2012.

**GNU.ORG** [Internet] O que é Software Livre. Disponível em: <http://www.gnu.org/philosophy/free-sw.pt-br.html>. Acessado em 19 de Maio de 2012

**HEXCEL**, Roberto A. (2002) Software Livre: Propostas de Ações de Governo para Incentivar o Uso de Software Livre, In: Relatório Técnico RT-DINF 004/2002. Universidade federal do Paraná, Brasil.

**KON**, Fabio; **LAGO**, Nelson; **MEIRELLES**, Paulo; **SABINO**, Vanessa. *Software Livre e Propriedade Intelectual: Aspectos Jurídicos, Licenças e Modelos de Negócio*

**LAGO**, Nelson. Licenças de Software Livre. Entrevista com Nelson Lago. Publicado na TV Serpro. Disponível em: [http://www.tv.serpro.gov.br/Entrevista\\_Nelson\\_Lago.ogv/view](http://www.tv.serpro.gov.br/Entrevista_Nelson_Lago.ogv/view). Acessado em 19 de Maio de 2012

**MURILLO**, Luis Felipe Rosado. Tecnologia, Política e Cultura na Comunidade Brasileira de Software Livre e de Código Aberto. 2009. Dissertação (Mestrado em Antropologia Social) Instituto de Filosofia e Ciências Humanas da Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, 2003.

**OPENSOURCE.ORG** [Internet]. The Open Source Definition. Disponível em: <http://www.opensource.org/docs/osd>. Acessado em 19 de março de 2012.

**PIRES**, Hindenburgo Francisco. Internet, Software Livre e Exclusão Digital: Impasses e opções no desenvolvimento de políticas públicas de alcance social no Brasil. Artigo publicado na Revista Geouery N° 12, Rio de Janeiro, pp.11-22, 2002. ISSN 1415-7543

**ROBOTOM REIS**, Christian. Caracterização de um processo de Software para Projetos de Software Livre. 2003. 247f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) Instituto de Matemática e de Computação da Universidade de São Paulo, São Paulo, 2003.

**SABINO**, Vanessa. *Um estudo sistemático de licenças de software livre*, 2011. 116f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2011.

**SABINO**, Vanessa; **KON**, Fabio. *Licenças de Software Livre - História e Características - Relatório Técnico RT-MAC-IME-USP 2009-01*.

**ELIAS**, Paulo César; **MATTOS**, Fernando Augusto M. *Informação e Software Livre no Capitalismo Contemporâneo*, 2007. 68f.

**SENN & MARIANO** Advogados Associados. ASPECTOS DO SOFTWARE LIVRE SOB O DIREITO PRIVADO. A proteção jurídica do software e a licença GNU/GPL.

**TAURION**, C. Software livre: potencialidades e modelos de negócio. Rio de Janeiro: Brasport, 2004. Publicado em 16/9/2006

**WIKIPEDIA** [Internet]. Software Livre. Disponível em: [http://pt.wikipedia.org/wiki/Software\\_Livre](http://pt.wikipedia.org/wiki/Software_Livre). Acessado em 15 de março de 2012.

**WIKIPEDIA** [Internet] GNU General Public License. Disponível em: [http://en.wikipedia.org/wiki/GNU\\_General\\_Public\\_License#Linking\\_and\\_derived\\_works](http://en.wikipedia.org/wiki/GNU_General_Public_License#Linking_and_derived_works). Acessado em 21 de Abril de 2012.

**GNU.ORG** [Internet] Questões Feitas Frequentemente (FAQ) sobre a GNU GPL. Disponível em: <http://www.gnu.org/licenses/gpl-faq.html#AssignCopyright>. Acessado em 26 de Março de 2012.

**OPENSOURCE.MIRRORING.DE** [Internet] Open Source Initiative OSI - Apache License, Version 2.0: Licensing. Disponível em: <http://opensource.mirroring.de/licenses/apache2.0.php>. Acessado em 21 de Abril de 2012.

**ROSENLAWS.COM** [Internet] Derivative Works. Disponível em: <http://www.rosenlaw.com/lj19.htm>. Acessado em 10 de Março de 2012.

**BLACKDUCKSOFTWARE.COM** [Internet] Learning Center. Disponível em: <http://osrc.blackducksoftware.com/learningcenter/index.php>. Acessado em 07 de Abril de 2012.

**REDHAT.COM** [Internet] Open Source from a Proprietary Perspective. Disponível em: [http://www.redhat.com/f/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan\\_OpenSourceProprietaryPersp.pdf](http://www.redhat.com/f/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan_OpenSourceProprietaryPersp.pdf). Acessado em: 21 de Março de 2012.