

Licenças de Software Livre História e Características*

Relatório Técnico RT-MAC-IME-USP 2009-01

Vanessa Sabino, Fabio Kon

Centro de Competência em Software Livre
Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

Março de 2009

1 Introdução

A questão das licenças é bastante importante para o desenvolvimento e adoção do software livre. O objetivo deste relatório é auxiliar pessoas que não são especialistas em licenciamento de software a entender melhor suas implicações, tornando o texto das principais licenças mais acessível e facilitando o processo de escolha de software realizado por gerentes, investidores, desenvolvedores, estudantes, cientistas e demais pessoas envolvidas com software.

Programas de software livre em geral são de fácil acesso. Porém, a simples obtenção de um programa não significa que a pessoa pode fazer o que quiser com ele. As licenças de software livre são documentos através dos quais os detentores dos direitos sobre um programa de computador autorizam usos de seu trabalho que, de outra forma, estariam protegidos pelas leis vigentes no local.

Além do uso como usuário final, esses usos autorizados permitem que desenvolvedores possam adaptar o software para necessidades mais específicas, utilizá-lo como fundação para construção de programas mais complexos, entre diversas outras possibilidades. Neste relatório, veremos como a escolha da licença influencia a forma como o software poderá ser usado, desenvolvido e distribuído.

Cada licença será discutida de forma sistemática, traduzindo as principais idéias contidas em seus termos para uma linguagem mais familiar ao desenvolvedor de software, explicando quando seu uso é recomendado e levantando as principais vantagens e desvan-

*Pesquisa apoiada pela CNPq e pelo Projeto Qualipso (*European Commission*).

tagens. Mas antes disso, de forma a contextualizar o estudo, será apresentado um breve histórico do movimento de software livre e suas principais características.

2 Breve Histórico do Software Livre

Com o surgimento dos primeiros computadores vendidos comercialmente, a partir da década de 1950, foram criados também os primeiros programas que iriam ser executados neles. Muitas vezes ocorria uma venda casada entre hardware e software, pois os programas eram fortemente acoplados à arquitetura das máquinas em que eram executados. Nessa época, o foco das empresas era na venda do hardware, e não eram colocadas muitas restrições no uso que as pessoas fariam do software [CK08]. Elas podiam adaptá-lo como quisessem, de forma a fazer melhor uso do hardware que tinham disponível, sem sofrer repreensões.

Na década de 1970 a situação começou a se modificar. Algumas empresas, como a Microsoft, não estavam satisfeitas com a forma como seus programas eram redistribuídos sem que a empresa recebesse *royalties* pelas cópias. Assim, em 3 de fevereiro de 1976, Bill Gates escreveu a *Open Letter to Hobbyists*, que foi publicada na *newsletter* do *Homebrew Computer Club* (Figura 1). Nessa carta, Bill Gates afirma que o total de *royalties* recebidos pelo Altair BASIC era equivalente a apenas dois dólares por hora gasta em seu desenvolvimento e documentação. Ele ainda alega que a prática de compartilhamento de software não é justa e afirma que tal prática evita que software bem feito seja escrito. Assim, nessa época começou uma mudança de postura na indústria, que passou a proibir que o software fosse copiado ou modificado. Surgiu então o que chamaremos de *software fechado*, caracterizado pelas restrições que são feitas à forma como ele será utilizado.

Como resposta a essa nova situação, surgiram iniciativas voltadas para retomar a liberdade de melhorar e compartilhar o software. Discutiremos a seguir duas delas: o Projeto GNU, combinado com a filosofia do Software Livre, e o movimento *Open Source* [Tau04].

2.1 O Projeto GNU e o Software Livre

Em 27 de setembro de 1983, Richard Stallman postou uma mensagem nos *newsgroups* *net.unix-wizards* e *net.usoft* com o assunto “*new Unix implementation*”. Nessa mensagem, ele informa que está começando a escrever um sistema compatível com UNIX chamado GNU (um acrônimo recursivo para *Gnu’s Not Unix*) e que ele será dado a todas as pessoas interessadas. Ele cita alguns componentes que seriam incluídos, tais como núcleo do sistema operacional, compilador C e editor de texto, e propõe algumas melhorias em relação aos sistemas UNIX existentes na época. Ele também explica na mensagem o motivo dele precisar escrever o GNU: segundo seus princípios, se ele gosta de um programa ele precisa compartilhá-lo com outras pessoas que também gostem dele. Para continuar usando computadores sem violar seus princípios, ele decidiu criar um conjunto suficiente de software livre para que ele pudesse prosseguir sem usar qualquer software que não fosse livre. Para finalizar a mensagem, ele pede contribuições na forma de máquinas, ajuda para escrever o software e dinheiro. No livro *Open Sources* [DOS99], Stallman conta um pouco mais sobre a história do projeto, cujos principais fatos serão discutidos aqui.

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates
Bill Gates
General Partner, Micro-Soft

Figura 1: Homebrew Computer Club Newsletter Volume 2, Issue 1

No início de 1984 Stallman largou seu emprego no laboratório de Inteligência Artificial do MIT, para garantir que ele teria os direitos sobre o trabalho, e começou a dedicar-se em tempo integral ao projeto. O primeiro programa criado foi o GNU Emacs, que foi disponibilizado por FTP. Porém, como naquela época muitas pessoas não tinham acesso à Internet, Stallman também começou a ganhar dinheiro vendendo cópias físicas do programa, em fita magnética, por 150 dólares. Ele considera esse negócio de distribuição de software livre um precursor das empresas que hoje distribuem o GNU/Linux. Pouco tempo depois foi desenvolvido o GCC (que na época significava *GNU C Compiler*, mas atualmente, com a adição de outras linguagens, é chamado de *GNU Compiler Collection*), que até hoje é um dos componentes mais importantes do sistema GNU. Ao longo dos anos, Stallman decidiu incorporar, ao sistema GNU, software que não foi escrito pelo projeto GNU, como por exemplo o Linux e o X Window System. Isso era possível pois esses programas também eram livres.

Enquanto o sistema GNU era desenvolvido, também foi sendo formado o conceito de *Free Software*, ou Software Livre, levando à criação da *Free Software Foundation* por Stallman em 1985. Segundo sua definição, um software é livre se:

1. Você tem a liberdade de executar o programa, para qualquer propósito.
2. Você tem a liberdade de modificar o programa para adaptá-lo às suas necessidades (para tornar essa liberdade efetiva na prática, você precisa ter acesso ao código fonte, já que fazer alterações em um programa sem ter o código fonte é muito difícil).
3. Você tem a liberdade de redistribuir cópias gratuitamente ou mediante pagamento.
4. Você tem a liberdade de distribuir versões modificadas do programa para que a comunidade possa se beneficiar de suas melhorias.

Como o objetivo do projeto GNU era garantir essas liberdades para os usuários, foi criado um sistema de distribuição chamado *copyleft*, que visava impedir que o software se tornasse fechado. Mais detalhes a respeito disso serão vistos na Seção 4.2.

2.2 O Movimento *Open Source*

Segundo Eric Raymond, devido ao caráter ético e político do movimento proposto pela *Free Software Foundation*, muitas empresas viam o software livre como “anti-capitalista” e eram relutantes em adotá-lo. Observando isso, ele teve a idéia de mudar a abordagem como seria apresentado o software livre para pessoas mais conservadoras, e criou o termo *Open Source* em 1997. Não usando a palavra *free*, Raymond estava não apenas evitando a confusão com gratuito, como também tirando a conotação esquerdista do termo proposto por Stallman.

Também em 1997, Bruce Perens havia escrito o *Debian Free Software Guidelines*, para definir o que seria aceito como software livre pela distribuição Debian, dado que havia outras licenças além daquelas propostas pela *Free Software Foundation* que alegavam serem livres. Então, Eric Raymond contatou Bruce Perens para discutir a idéia de *Open Source* e decidiram a partir daí adaptar o documento *Debian Free Software Guidelines*

para formar a *Open Source Definition*, ou Definição de Código Aberto. Eles registraram a marca *Open Source* e formaram a *Open Source Initiative* (OSI) [DOS99].

No início de 1998 tivemos o primeiro caso de uma empresa já consolidada no mercado abrir o código de seu software: o Netscape. Junto com Eric Raymond, os executivos da Netscape escreveram uma licença que adotava alguns princípios de *copyleft*, mas que permitia à Netscape continuar distribuindo versões fechadas com o código aberto do navegador. Em seguida, Raymond publicou um pedido à comunidade intitulado *Goodbye, "Free Software"; Hello, "Open Source"*, em que insistia que o termo *open source* era melhor do que *free software* e devia ser adotado.

Com essa nova abordagem, que ressaltava os benefícios técnicos decorrentes da metodologia adotada pela comunidade, e tendo como exemplo a Netscape, a adoção do software livre por parte das empresas sofreu um grande impulso. Porém, a maior disseminação do software livre nesses termos não deixou Stallman satisfeito. Segundo ele, se as pessoas não introjetam a liberdade associada ao software livre, elas voltarão a usar software fechado quando este apresentar vantagens práticas.

Stallman também argumenta que a expressão “código aberto” tem como significado óbvio simplesmente “você pode olhar para o código”, o que é um critério muito mais fraco do que a definição oficial de código aberto, que pode ser vista a seguir.

2.2.1 Definição de Código Aberto

Introdução

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

1. Redistribuição Livre

A licença não deve restringir qualquer das partes de vender ou doar o software como um componente de uma distribuição agregada de software, contendo programas oriundos de várias fontes diferentes. A licença não deve exigir *royalties* ou qualquer outro tipo de pagamento para venda.

2. Código Fonte

O programa deve incluir o código fonte, e deve permitir a distribuição na forma de código fonte bem como na forma compilada. Quando alguma forma do produto não é distribuída com o código fonte, é necessário haver meios bem divulgados para obtenção do código por não mais que um custo razoável de reprodução, preferencialmente através de download pela Internet gratuitamente. O código fonte deve ser a forma preferencial pela qual um programador alteraria o programa. Código fonte obscurecido deliberadamente não é permitido. Formas intermediárias, como a saída de um processador ou tradutor, não são permitidas.

3. Trabalhos Derivados

A licença deve permitir modificações e trabalhos derivados e precisa permitir que eles sejam distribuídos sob os mesmos termos da licença do software original.

4. Integridade do Código Fonte do Autor

A licença pode restringir a distribuição de código fonte em forma modificada somente se a licença permitir a distribuição de “arquivos de *patch*” com o código fonte para o propósito de modificar o programa em tempo de compilação. A licença deve permitir explicitamente a distribuição do software compilado a partir de um código modificado. A licença pode exigir que trabalhos derivados usem um nome ou número de versão diferentes do original.

5. Sem Discriminação a Pessoas ou Grupos

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

6. Sem Discriminação a Áreas de Empreendimento

A licença não deve restringir qualquer pessoa a fazer uso do programa em uma área de empreendimento específica. Por exemplo, ela não pode restringir o uso do programa comercialmente ou o uso em pesquisas genéticas.

7. Distribuição da Licença

Os direitos associados ao programa devem ser aplicáveis a todos para quem o programa é redistribuído, sem a necessidade de execução de licenças adicionais para essas partes.

8. A Licença Não Deve Ser Específica a um Produto

Os direitos associados ao programa não devem depender dele ser parte de uma distribuição específica de software. Caso o programa seja extraído dessa distribuição e usado ou distribuído nos termos da licença do programa, todas as partes para as quais o programa é redistribuído devem ter os mesmos direitos que aqueles concedidos em conjunto com a distribuição de software original.

9. A Licença Não Deve Restringir Outro Software

A licença não deve colocar restrições em outro software que seja distribuído junto com o software licenciado. Por exemplo, a licença não deve exigir que todos outros programas distribuídos no mesmo meio sejam software de código aberto.

10. A Licença Deve Ser Neutra às Tecnologias

Nenhuma condição da licença deve ser estabelecida em uma tecnologia individual específica ou estilo de interface.

(Tradução livre da *Open Source Definition* publicada em opensource.org/docs/osd)

Baseando-se na Definição de Código Aberto, a OSI aprova as licenças que podem ser consideradas *open source*. Para isso, as licenças passam por um processo público de revisão para assegurar que estão em conformidade com as normas e expectativas da comunidade. A OSI conta com uma lista de mais de 70 licenças aprovadas (www.opensource.org/licenses) e possui um comitê para tratar do assunto de proliferação das licenças, visando criar mecanismos para facilitar a escolha da licença.

3 A Importância do Software Livre

Estima-se que hoje há centenas de milhões de usuários de software livre no mundo¹. Se considerarmos também usuários indiretos, que usam serviços baseados em software livre, como Google, Amazon ou eBay, esse número é ainda maior. Neste capítulo, veremos as principais vantagens e desvantagens deste modelo, além das perspectivas de longo prazo.

3.1 Vantagens do Software Livre

A *Free Software Foundation* e a *Open Source Initiative* apresentam justificativas diferentes para o uso do software livre. Neste relatório, focaremos no discurso da *Open Source Initiative*, não discutindo em profundidade as questões éticas relacionadas ao software livre que são defendidas pela *Free Software Foundation*.

A principal vantagem do software livre é permitir o compartilhamento do código fonte. Como consequência desse compartilhamento, evita-se a duplicação de esforços quando mais de uma entidade está interessada no desenvolvimento de uma aplicação com funcionalidade similares, reduzindo assim o custo do desenvolvimento.

Além disso, autores como Eric Raymond [Ray01] afirmam que software livre tem condições de ter maior qualidade do que seus equivalentes fechados. Uma das justificativas para essa afirmação de Raymond é conhecida como “A Lei de Linus”, que diz “dados olhos suficientes, todos os *bugs* são superficiais”. Isso significa que, com o maior número de usuários que tem acesso ao programa e até ao código-fonte, o software é testado melhor e os problemas existentes no código são encontrados mais rapidamente. Outro fator que contribui para a qualidade é o orgulho pessoal do desenvolvedor, pois a partir do momento que seu código poderá ser lido por mais pessoas é esperado que ele seja mais cuidadoso com seu trabalho. A competição também é facilitada no software livre e, assim, se o grupo original de desenvolvedores não está fazendo um bom trabalho para manter o projeto, é possível que um novo grupo faça um *fork* para suprir as deficiências.

No Brasil, caracterizado por um mercado local, onde apenas uma parte inexpressiva da indústria de software usa o modelo de desenvolvimento de software fechado para venda do produto sem customizações, o software livre pode trazer ainda mais benefícios. Ao desenvolver serviços e soluções baseadas em software livre, as empresas brasileiras podem deixar de investir em licenças pagas promovendo a evasão de divisas e contribuindo negativamente para a balança comercial brasileira.

Para os usuários também é vantagem o software livre, pois evita a dependência de um fornecedor. Isso traz tanto uma vantagem financeira, dado que normalmente é necessário pagar por novas versões do sistema quando o software é fechado, como também maior liberdade para o usuário, que pode adaptar o software para suas necessidades. É possível corrigir falhas de segurança e *bugs*, escrever uma documentação melhor ou contratar uma empresa que faça isso independentemente de quem seja o autor original [Web04]. Além disso, se o fornecedor original abandona o projeto, no caso do software fechado não há nada que possa ser feito para continuar o desenvolvimento do projeto, enquanto que no software livre é possível que outro grupo adote o projeto e continue a evoluir o código.

¹Para essa estimativa escolhemos o navegador Firefox, que tem um *market-share* de 22% sobre o número total de usuários de Internet, que é de aproximadamente 1,5 bilhão.

3.2 Desvantagens do Software Livre

Também podemos levantar algumas desvantagens do software livre em relação a alternativas fechadas. Um dos principais motivos que leva uma empresa a optar por um software fechado quando há um similar livre disponível é a ausência de garantias e suporte desse último. As licenças de software livre em geral eximem o autor de qualquer responsabilidade tanto quanto é permitido pelas leis do local. Dessa forma, em casos em que a empresa precisa fornecer garantias aos seus clientes, ou quando a indisponibilidade de um sistema pode causar grandes prejuízos, pode ser melhor que a empresa adquira uma solução em que eventuais problemas sejam delegados a um fornecedor ou que esse tenha que indenizar a empresa. Porém, é importante deixar claro que, apesar das licenças de software livre normalmente incluírem cláusulas sobre a ausência de responsabilidades, há casos em que empresas optam por fornecer, como um serviço, garantias e/ou suporte para um determinado software livre. Além disso, grande parte do software fechado disponível também busca em seu contrato se eximir de responsabilidades tanto quanto a legislação permite.

Qualidade, reputação e imagem também são vistos como desvantagens na adoção do software livre. Quando não há uma empresa de renome por trás do software oferecido, há uma maior dificuldade em avaliar as alternativas, além de um receio de que o produto seja abandonado e deixe-se de oferecer suporte para ele. Também influi negativamente na sua imagem o fato do software estar disponível gratuitamente.

Já do ponto de vista de quem produz software, optar pelo modelo aberto pode ser visto como desvantagem à medida que a propriedade intelectual está exposta. Como os concorrentes têm fácil acesso ao código fonte, é necessário que a empresa criadora do software mantenha seu diferencial para garantir seu mercado a longo prazo. Caso contrário, o software pode tornar-se um *commodity* e as possibilidades de lucro da empresa são reduzidas [Web04]. Por outro lado, no caso de uma licença que obriga que as melhorias sejam disponibilizadas como software livre, é mais difícil que uma nova empresa adquira vantagem competitiva sobre outra que já está consolidada no mercado, pois qualquer diferencial no produto pode ser prontamente absorvido pela empresa líder [DOS99].

Observa-se que o modelo de negócio tradicional de vender software da mesma forma como se vende qualquer outro produto não funciona bem. É necessário buscar outros modelos de negócio.

3.3 Perspectivas do Software Livre

O modelo do software livre possui características bastante interessantes para o mercado de Tecnologia da Informação. Há uma certa ambigüidade na relação entre clientes, empresas e a comunidade de desenvolvimento de software, permitindo novos modelos de negócio. Os consumidores têm a possibilidade de também contribuir para o bem coletivo, agregando valor. Porém, a parcela de consumidores que realmente contribui é muito pequena, e os demais requerem uma quantidade desproporcional de serviços de suporte, que não são facilmente escaláveis. A competição com produtos fechados, que já arrecadam dinheiro com a venda do software, empurra para baixo o valor do suporte, dificultando a situação das empresas que têm como modelo a venda de serviços baseados em software livre [Web04].

Em um recente estudo da Forrester [For08] sobre o uso de software livre e seu impacto na indústria de software, foram observadas as seguintes tendências:

- crescimento da adoção de software livre pelo usuário final, na forma de ferramentas de produtividade e aplicações de negócio;
- diferenças na adoção de software livre de acordo com ramo de atividade, com fábricas adotando fortemente o software livre em sua infraestrutura enquanto serviços financeiros usam o software livre em aplicações de mais alto nível;
- uso de software livre em aplicações de missão crítica, serviços e produtos;
- altos índices de satisfação em relação a custo e qualidade do software livre;
- diversidade de critérios como motivadores para adoção de software livre, destacando custo, independência, flexibilidade e inovação;
- adoção, por parte das empresas, das boas práticas e princípios da comunidade de software livre;
- crescimento de provedores de serviço e centros de competência para prover suporte comercial para produtos de software livre.

Hoje em dia o software livre aparece na maior parte das projeções sobre o futuro do software. Porém, é importante ressaltar que cada novo paradigma que surge na indústria altera substancialmente as perspectivas, tornando muito difícil prever como estará o mercado de software daqui a alguns anos. [CK08].

4 Levantamento e Classificação de Licenças

O detentor dos direitos patrimoniais sobre um software, quando decide torná-lo livre, deve escolher os termos em que seu trabalho será distribuído, ou seja, os direitos que ele estará transferindo para as outras pessoas e quais as condições que serão aplicadas. O documento que formaliza esse ato é a licença, que normalmente é distribuída junto com o código fonte.

Há inúmeras possibilidades para redigir o texto de uma licença de software livre, mas a prática mais comum é reaproveitar alguma das licenças já consolidadas na comunidade. Dessa forma, reduz-se a proliferação de licenças, que deve ser evitada pois gera trabalho adicional para os usuários, uma vez que torna-se necessário para eles estudar os termos de cada nova licença presente no software que irão utilizar. No portal SourceForge, que hospeda um grande número de projetos de código aberto, ao criar um novo projeto são apresentadas oito opções de licenças, que são as mais utilizadas naquele repositório, além da opção de domínio público:

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

- MIT License
- Apache License V2.0
- Artistic License
- Mozilla Public License 1.1 (MPL 1.1)
- Academic Free License (AFL)

Além disso, no final da página que lista as licenças, há um ponteiro para visualização de todas as licenças disponíveis, contendo, em outubro de 2008, 74 opções. Há ainda a possibilidade do usuário incluir sua própria licença.

A *Open Source Initiative* (OSI) mantém uma lista de licenças aprovadas de acordo com a definição de código aberto (vide Seção 2.2), organizadas em diversas categorias, além de uma lista de licenças atualmente buscando aprovação. A OSI também está trabalhando com um comitê (anti-)proliferação de licenças, visando separar e dar destaque a algumas licenças que passarão a ser *recomendadas*, em oposição às meramente *aprovadas*.

Vamos discutir neste relatório algumas das principais licenças utilizadas atualmente pela comunidade, visando seu melhor entendimento e aplicação. Considerando que uma das questões mais importantes é a compatibilidade entre licenças, vamos separar as licenças descritas aqui em três categorias, de acordo com a presença de termos que impõem restrições de licenciamento na redistribuição do trabalho ou criação de trabalhos derivados. Em relação a esta característica, as licenças são consideradas *permissivas* ou *recíprocas*, sendo que entre as recíprocas devemos ainda considerar que algumas forçam que seja mantida a mesma licença em mais casos do que outras, e assim as dividimos entre recíprocas parciais, que também recebem a denominação de *copyleft* fraco, e recíprocas totais.

4.1 Permissivas

As licenças permissivas, também chamadas de licenças acadêmicas por alguns autores, como Rosen [Ros05] e Laurent [Lau04], em referência às origens das licenças BSD (*University of California, Berkeley*) e MIT (*Massachusetts Institute of Technology*), impõem poucas restrições às pessoas que obtém o produto. Muitas vezes, tais licenças são usadas em projetos de pesquisa de universidades, que servem como prova de conceito de alguma tecnologia que poderá ser explorada comercialmente no futuro. No caso das licenças permissivas, não é feita nenhuma restrição ao licenciamento de trabalhos derivados, estes podendo inclusive serem distribuídos sob uma licença fechada. Nesta seção serão discutidas as licenças BSD, MIT e Apache.

Licenças permissivas são uma ótima opção para projetos cujo objetivo é atingir o maior número de pessoas, não importando se na forma de software livre ou de software fechado. Temos vários exemplos desse modelo no BSD Unix, que continha o software de TCP/IP que hoje é usado na maior parte das implementações desse protocolo, incluindo a da Microsoft [Lau04]. Outro exemplo é o BIND, Berkeley Internet Name Daemon, cuja implementação livre é até hoje usada nos principais servidores de DNS, apesar de haver também várias implementações fechadas. Segundo Laurent [Lau04], há bilhões de

dólares em atividade econômica associada apenas com a pilha de software para Internet originalmente liberada sob a licença BSD.

Alguns argumentam que o uso desse tipo de licença não incentiva o modelo de software livre, pois empresas se aproveitam da comunidade para desenvolver software que será fechado. Porém, quando são usadas licenças permissivas, em geral os autores estão cientes dessa possibilidade e não vêem isso como um problema. Um caso conhecido em que, de fato, os autores se arrependeram da licença que adotaram é o do Kerberos, desenvolvido no MIT, que posteriormente foi adotado pela Microsoft, que desenvolveu extensões fechadas [Lau04]. Mas o mais provável, caso a licença não permitisse isso, seria que a Microsoft adotasse algum outro sistema de segurança e o Kerberos não se tornaria tão popular.

Por outro lado, em alguns casos, não é necessário que haja restrições na licença para garantir a continuidade do modelo de desenvolvimento aberto, como no exemplo do servidor Apache. Duas características explicam o domínio do servidor Web livre no mercado, segundo Laurent [Lau04]: a marca forte, cujo uso é protegido pela própria licença do Apache, e a importância da conformidade com os padrões, que evita a disseminação de extensões fechadas.

4.1.1 A Licença BSD

A primeira licença que iremos estudar é a BSD (www.creativecommons.org/licenses/BSD/legalcode), que foi a primeira licença de software livre escrita e até hoje é uma das mais usadas. Criada originalmente pela Universidade da Califórnia em Berkeley para seu sistema operacional derivado do UNIX chamado Berkeley Software Distribution, a licença BSD é usada como modelo por uma ampla gama de software licenciado 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 ciclo virtuoso em que mais comunidades a adotaram, tornando-a ainda mais reconhecida.

Porém, originalmente, a licença BSD possuía uma cláusula que determinava que todo material de divulgação relacionado ao software precisava conter uma afirmação que dizia “este produto inclui software desenvolvido pela Universidade da Califórnia, Berkeley e seus contribuidores”. Tal cláusula, que ficou conhecida como “cláusula de propaganda da licença BSD”, não causava muito problema quando era usada apenas no seu contexto original, porém, à medida que outras pessoas e organizações passaram a adotar também a licença BSD, essa cláusula era adaptada para conter outros nomes, e assim surgiam projetos que integravam diversos componentes e determinavam uma longa lista de frases a serem incluídas quando se falava sobre o projeto. Como consequência, em 1999, tal cláusula foi removida, originando a “Licença BSD Simplificada”, também conhecida como “Nova Licença BSD” ou “Versão de 3 cláusulas da Licença BSD”.

Assim, na versão atualmente utilizada da licença BSD, temos os seguintes termos, apresentados em tradução realizada pela Escola de Direito da Fundação Getúlio Vargas:

Direitos autorais (C) <ANO>, <TITULAR>

Todos os direitos reservados.

A redistribuição e o uso nas formas binária e código fonte, com ou sem modificações, são permitidos contanto que as condições abaixo sejam cumpridas:

- Redistribuições do código fonte devem conter o aviso de direitos autorais acima, esta lista de condições e o aviso de isenção de garantias subsequente.
- Redistribuições na forma binária devem reproduzir o aviso de direitos autorais acima, esta lista de condições e o aviso de isenção de garantias subsequente na documentação e/ou materiais fornecidos com a distribuição.
- Nem o nome da <ORGANIZAÇÃO> nem o nome dos contribuidores podem ser utilizados para endossar ou promover produtos derivados deste software sem autorização prévia específica por escrito.

Apesar dessa licença utilizar uma linguagem leiga, cujos termos não necessariamente correspondem aos termos encontrados nas leis que visam proteger o autor do software, através deste documento o detentor dos direitos autorais permite que outras pessoas usem, modifiquem e distribuam o software. As únicas exigências são que o nome do autor original não seja utilizado em trabalhos derivados sem permissão, visando proteger sua reputação, dado que o autor pode não ter relação alguma com as modificações realizadas, e no caso de redistribuição do código fonte ou binário, modificado ou não, é necessário que seja mencionado o copyright original e os termos da licença.

É importante ressaltar que a exigência de reproduzir a lista de condições para redistribuição e o aviso legal de isenção de garantias não significa que o trabalho sendo redistribuído precisa estar sob a mesma licença. Além disso, é possível redistribuir o binário sem fornecer o código fonte. Basta que conste na documentação que foi utilizado o software em questão, e que ele foi licenciado nos termos descritos acima.

No final da licença há o aviso legal sobre garantias e responsabilidades apresentado abaixo, que visa proteger o autor de ser processado judicialmente por qualquer insatisfação causada em consequência do uso do software. Porém, tal termo está subordinado às leis locais, que no caso brasileiro, dependendo da relação estabelecida entre o fornecedor e o cliente, não permitem a distribuição de um software com ausência total de garantias.

ESTE SOFTWARE É FORNECIDO PELOS DETENTORES DE DIREITOS AUTORAIS E CONTRIBUIDORES "COMO ESTÁ", ISENTO DE GARANTIAS EXPRESSAS OU TÁCITAS, INCLUINDO, SEM LIMITAÇÃO, QUAISQUER GARANTIAS IMPLÍCITAS DE COMERCIALIZABILIDADE OU DE ADEQUAÇÃO A FINALIDADES ESPECÍFICAS. EM NENHUMA HIPÓTESE OS TITULARES DE DIREITOS AUTORAIS E CONTRIBUIDORES SERÃO RESPONSÁVEIS POR QUAISQUER DANOS, DIRETOS, INDIRETOS, INCIDENTAIS, ESPECIAIS, EXEMPLARES OU CONSEQUENTES, (INCLUINDO, SEM LIMITAÇÃO, FORNECIMENTO DE BENS OU SERVIÇOS SUBSTITUTOS, PERDA DE USO OU DADOS, LUCROS CESSANTES, OU INTERRUPTÃO DE ATIVIDADES), CAUSADOS POR QUAISQUER MOTIVOS E SOB QUALQUER TEORIA DE RESPONSABILIDADE, SEJA RESPONSABILIDADE CONTRATUAL, RESTRITA, ILÍCITO CIVIL, OU QUALQUER OUTRA, COMO DECORRÊNCIA DE USO DESTES SOFTWARE, MESMO QUE HOUVESSEM SIDO AVISADOS DA POSSIBILIDADE DE TAIS DANOS.

Por fim, há também versões da licença BSD que removem a terceira condição de redistribuição, relacionada ao autor não endossar trabalhos derivados, restando apenas as duas cláusulas que requerem a reprodução do *copyright* e condições na redistribuição.

4.1.2 Vantagens e Desvantagens

Uma das principais desvantagens da licença BSD 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. Sempre é necessário verificar com mais cuidado, no texto da licença, se o software em questão está licenciado pela BSD original, a simplificada, ou a versão sem endosso. Outra desvantagem, do ponto de vista jurídico, é que seus termos são um tanto vagos e não são mapeados diretamente aos termos utilizados no licenciamento de software, havendo um maior esforço de interpretação da licença para demonstrar que, de fato, a maioria dos direitos em geral restritos ao autor estão sendo transferidos para o licenciado. Por outro lado, temos como vantagens a simplicidade da licença e sua ampla adoção, que facilitam seu entendimento pelo público geral.

Portanto, o exposto acima faz da licença BSD uma excelente escolha para projetos em que não há uma preocupação a respeito de quais serão os termos que outras pessoas usarão ao redistribuir o trabalho original ou derivado. Um software distribuído sob a licença BSD se aproxima bastante, em relação a seus direitos intrínsecos, de um software em domínio público.

4.1.3 A Licença MIT/X11

A Licença MIT (www.opensource.org/licenses/mit-license.php), criada pelo *Massachusetts Institute of Technology*, é também conhecida como Licença X11 ou X, por ter sido redigida para o X Window System, desenvolvido no MIT em 1987.

Essa também é uma licença permissiva e é considerada equivalente à BSD Simplificada sem a cláusula de endosso. Porém, seu texto é bem mais explícito ao tratar dos direitos que estão sendo transferidos, afirmando que qualquer pessoa que obtém uma cópia do software e seus arquivos de documentação associados pode lidar com eles sem restrição, incluindo sem limitação os direitos a usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do software. As condições impostas para tanto são ape-

nas manter o aviso de *copyright* e uma cópia da licença em todas as cópias ou porções substanciais do software.

Para finalizar, a licença contém uma cláusula sobre a ausência de garantias e responsabilidades bastante similar à da licença BSD, protegendo os detentores do direito autoral de qualquer processo judicial relacionado ao software. Nesta licença ainda é excluída explicitamente a responsabilidade de não infração, que pode ocorrer, por exemplo, quando alguém usa a propriedade intelectual de outra pessoa sem a devida autorização. Isso cobre casos em que o autor do software, acidentalmente ou não, tenha utilizado algum material protegido sob direitos autorais ou uma idéia patenteada sem obter uma licença para tanto, o que pode gerar um processo contra o autor, os distribuidores do software e até seus usuários. Porém, assim como explicado anteriormente, a ausência de responsabilidades tem sua validade limitada pelas leis vigentes.

4.1.4 Vantagens e Desvantagens

Essa licença é a recomendada pela *Free Software Foundation* quando se busca uma licença permissiva, pois é bastante conhecida e, ao contrário da BSD, não possui múltiplas versões com cláusulas que podem gerar dificuldades adicionais, tais como a cláusula de propaganda da BSD que pode gerar incompatibilidades com outras licenças. Outra vantagem da MIT em relação à BSD é a maior clareza dos seus termos ao declarar explicitamente que é permitido, por exemplo, sublicenciar ou vender cópias do software, que são direitos apenas implícitos na BSD. A questão do sublicenciamento é bastante importante quando o software será usado como parte de um trabalho coletivo ou derivado que será distribuído sob outra licença, o que é bastante comum nas práticas de software livre. Se não houver o direito de sublicenciamento, então apenas o detedor do direito autoral pode conceder a licença. Desta forma, o usuário de um trabalho derivado precisa obter a licença tanto do autor desse trabalho como também dos detentores dos direitos de cada componente que faz parte dele, sendo necessário identificar todos esses componentes e pessoas envolvidas, aumentando a dificuldade, os riscos e a complexidade jurídica. Por outro lado, quando é permitido o sublicenciamento, a pessoa que está emitindo a licença, que valerá para todos os componentes de seu software, fica responsável por analisar os termos de cada um dos componentes e certificar-se da compatibilidade entre as licenças. Assim, ao chegar no usuário final, o potencial de problemas é restrito a uma única licença.

4.1.5 A Licença Apache

A Licença Apache está atualmente na versão 2.0 (www.apache.org/licenses/LICENSE-2.0.html) e é usada por um dos projetos mais conhecidos de software livre, o servidor Web Apache, assim como pela maior parte dos outros projetos pertencentes à Fundação Apache, além de projetos independentes que optaram por usar essa licença.

A Apache também é uma licença permissiva e, na versão 1.1, seu texto era bastante similar ao da BSD, porém dando ênfase à proteção da marca Apache. Havia uma cláusula similar à cláusula de propaganda da BSD, obrigando que a documentação ou o software, quando redistribuídos, incluíssem a frase “este produto inclui software desenvolvido pela Apache Software Foundation (<http://www.apache.org/>)” e duas cláusulas proibindo o uso do nome Apache sem permissão escrita prévia, tanto para endossar trabalhos derivados, como na BSD, como também o uso como parte do nome do produto. Em 2004, a licença

foi totalmente reescrita e seu texto ficou bem mais longo e complexo, detalhando melhor os direitos concedidos, conforme será visto a seguir.

A primeira parte da licença contém as definições de palavras-chave que serão utilizadas no decorrer da licença, tais como “Entidade Legal”, “Você” e “Fonte”. A definição de fonte é mais abrangente do que a convencional, englobando não apenas o código fonte do software, como também fonte da documentação e arquivos de configuração. Analogamente, na definição de “Objeto” é incluída qualquer coisa resultante da transformação ou tradução mecânica da fonte, tais como código compilado, documentação e conversões para outros tipos de mídia. Essa licença também define o que é um “Trabalho Derivado”, excluindo explicitamente trabalhos que se mantêm separados do trabalho sendo licenciado ou que meramente são ligados à sua interface. A próxima definição é a de “Contribuição”, que é uma modificação do trabalho original que será incluída no trabalho em futuras versões. Como será visto adiante, a contribuição também pode ser licenciada nos termos da licença Apache.

As próximas duas cláusulas se referem à concessão de direitos autorais e patentes. Quanto aos direitos autorais, é dada uma licença irrevogável para reproduzir, preparar trabalhos derivados, mostrar publicamente, sublicenciar e distribuir o trabalho e seus derivados, na forma de fonte ou objeto, sujeitos aos termos e condições da licença. Essa cláusula enumera os direitos protegidos pela lei que estão sendo concedidos à pessoa que obtém a licença, adaptando-se bem às leis americanas, porém no caso brasileiro, em que a legislação de direitos autorais é muito mais restritiva, acaba deixando de lado alguns direitos fundamentais, como por exemplo o do uso do software. Nesse sentido, a versão anterior da licença era mais apropriada, já que definia os direitos concedidos de forma mais abrangente e incluía em seu texto a permissão de uso.

Em seguida, são colocadas algumas condições para a redistribuição do trabalho e seus derivados:

- incluir uma cópia da licença;
- incluir avisos em todos os arquivos modificados informando sobre a alteração;
- manter na fonte de trabalhos derivados todos os avisos de direitos autorais, patentes e marcas registradas que são pertinentes;
- se o trabalho incluir um arquivo texto chamado “NOTICE”, então qualquer trabalho derivado distribuído deve incluir os avisos pertinentes contidos nesse arquivo da forma como está detalhado na licença.

A cláusula sobre redistribuição termina informando explicitamente que é permitido licenciar sob outros termos qualquer modificação ou trabalho derivado, desde que o uso, reprodução e distribuição do trabalho que foi obtido pela licença Apache esteja de acordo com seus termos.

Já no caso das contribuições enviadas, que são tratadas na cláusula seguinte, fica determinado que é usada a princípio a licença Apache, sem termos ou condições adicionais, a não ser que tenha sido realizado algum outro tipo de acordo de licenciamento entre as partes.

As quatro últimas cláusulas tratam das marcas registradas e responsabilidades legais. A licença limita-se a permitir o uso da marca apenas para uso razoável para descrever a

origem do trabalho e reproduzir o arquivo NOTICE. Dessa forma, fica implícito que não é permitido associar os nomes relacionados ao trabalho original a trabalhos derivados além do necessário para indicar a origem do trabalho. A cláusula que informa sobre a falta de garantias e responsabilidade é bastante similar àquela vista anteriormente na licença MIT, assim como a cláusula sobre limitações da responsabilidade. A principal diferença está na proteção das contribuições e seus responsáveis, já que esses também são tratados no resto da licença. Na última cláusula, é tratada a questão de outras entidades aceitarem prover garantias ou serem responsabilizadas pelo trabalho, o que é permitido desde que fique claro que a entidade está agindo por conta própria e ficará totalmente responsável pelas ações judiciais decorrentes, de forma a não interferir nas proteções judiciais que recaem sobre os contribuidores de acordo com as cláusulas anteriores da licença.

Após apresentar os termos acima, a licença inclui instruções sobre como aplicá-la a um trabalho. Como o texto da licença é bastante extenso, tornando inconveniente apresentá-lo por completo em cada arquivo do código fonte do projeto, a licença dá como diretriz que seja usado o seguinte texto:

```
Copyright [ano] [nome do detentor dos direitos]

Licensed under the Apache License, Version 2.0 (the "License"); you may
not use this file except in compliance with the License. You may obtain a
copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an 'AS IS' BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

4.1.6 Vantagens e Desvantagens

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. Em particular, o fato da licença Apache deixar explícito na cláusula 4, 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. Outra vantagem dessa licença é o tratamento de contribuições, tornando mais transparente a incorporação das mesmas em um projeto quando é feita a opção por manter a mesma forma de licenciamento. Para formalizar o processo de contribuições, foi criado o *Apache Contributor License Agreement*, que transmite à *Apache Software Foundation* todos os direitos de propriedade intelectual necessários para que a fundação possa licenciar a contribuição, o que é importante caso ocorra a decisão de uma alteração de licença do projeto. O fato do texto da licença ser mais extenso do que os vistos anteriormente é visto como uma desvantagem por algumas pessoas, porém o ganho em precisão compensa esse fato. O principal problema da Apache é o fato de sua compatibilidade com a GPL 2.0 ser discutível, dado que ela impõe algumas restrições que não estão no texto dessa versão da GPL.

4.2 Recíprocas Totais

As licenças recíprocas totais determinam que qualquer trabalho derivado precisa ser distribuído sob os mesmos termos da licença original. Isso também é chamado de *copyleft*, um termo criado pela *Free Software Foundation* (www.gnu.org/copyleft). A idéia do *copyleft* é dar permissão a todos para executar, copiar, modificar e distribuir versões modificadas do programa, mas impedir que sejam adicionadas restrições a essas versões redistribuídas. Tal idéia visa fortalecer o software livre como um todo, não permitindo que melhorias do software sejam retiradas do alcance da comunidade. O resultado esperado é que a quantidade de software livre aumente cada vez mais, beneficiando todos os envolvidos na cadeia produtiva do software livre. Além disso, a reciprocidade contribui para manter a compatibilidade entre diversas versões de um determinado sistema, dado que quando novas funcionalidades são feitas de forma fechada, fica mais difícil replicá-las nas diferentes versões. Por outro lado, tal abordagem também sofre críticas de dentro da comunidade, pois o software licenciado nesse modelo acaba ficando de certa forma isolado dos demais devido a incompatibilidades nas licenças. Na prática, software licenciado sob o modelo permissivo, em geral, pode ser incorporado em software licenciado como recíproco, já que licenças permissivas permitem a redistribuição sob outros termos, inclusive os de licenças recíprocas. Porém, o inverso não é verdadeiro e, assim, software sob licenças recíprocas não pode ser utilizado em vários projetos de software livre que usam alguma outra licença.

A licença que deu origem à idéia de *copyleft* foi a *General Public License*, ou GPL (www.gnu.org/licenses/gpl.html), da *Free Software Foundation*. Nesta seção iremos estudar algumas de suas versões: a GPL 2.0, GPLv3 e *Affero General Public License*, ou AGPL.

4.2.1 GPL 2.0

A licença GPL (<http://www.gnu.org/licenses/old-licenses/gpl-1.0.html>) foi escrita em 1989 pela *Free Software Foundation*. Dois anos depois, em junho de 1991, foram feitas pequenas modificações na licença, gerando a versão 2.0 (<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>). Essa versão manteve-se até 2007, quando saiu a GPLv3, que será vista na próxima seção. Devido ao grande número de projetos licenciados sob a versão 2.0, incluiremos neste relatório uma descrição detalhada dessa versão da licença.

A GPL 2.0 inclui um preâmbulo que explica os princípios que baseiam a licença e seus principais objetivos. Apesar desse preâmbulo ser bastante citado nas discussões, ele não tem valor jurídico, ou seja, por não fazer parte dos termos e condições, suas palavras não precisam ser obedecidas por quem obtém a licença do software. Seu objetivo é apenas melhorar o entendimento da GPL em seu contexto [Ros05], explicando o que é software livre e a importância do *copyleft*.

A licença GPL pode ser copiada, distribuída e aplicada a qualquer software cujo detentor dos direitos autorais assim desejar. Porém, diferentemente de outras licenças, como a BSD, o texto da GPL não pode ser alterado sem autorização, ou seja, não é permitido que seja feita uma licença derivada dela. No final da licença, há uma explicação sobre como aplicá-la a um trabalho. A *Free Software Foundation* recomenda que o autor que usa a GPL permita que seu trabalho esteja licenciado sob a versão mais recente

da licença ou qualquer versão posterior, de forma que quando surgir uma nova versão o usuário da licença possa escolher qual das versões estará utilizando. Dessa forma, evita-se incompatibilidade entre programas mais antigos e mais novos que optaram por utilizar a GPL. Porém, muitas pessoas preferem ter maior controle sobre quais são os termos em que seu software está licenciado e, assim, não deixam aberto para qualquer versão posterior criada pela *Free Software Foundation*. Um exemplo importante dessa escolha é o núcleo do Linux.

A Seção 0 da GPL define ao que ela se aplica, que seria qualquer programa ou trabalho que contém uma nota afirmando que está sendo licenciado pelos termos da GPL. No resto da licença, esse objeto a ser licenciado é chamado simplesmente de *Programa*. A Seção 0 também afirma que a licença se aplica a qualquer trabalho derivado segundo a lei de direitos autorais, porém ao mesmo tempo define trabalho derivado como “um trabalho contendo o Programa ou parte dele, literalmente ou com modificações e/ou traduzido para outra língua” (GNU General Public License, version 2, 1991). Essa definição dada a trabalho derivado difere tanto da legislação americana quanto da brasileira. Segundo a definição dada pela GPL, mesmo trabalhos coletivos são considerados como trabalhos derivados. Tal definição é de importância fundamental na aplicação da GPL e motivo de muita controvérsia quanto à necessidade de reciprocidade em diferentes usos do software. Além disso, a Seção 0 afirma que o ato de executar o Programa não é restrito de nenhuma forma e que a saída do Programa só está coberta pela licença se seu conteúdo constituir de alguma forma um trabalho baseado no Programa, o que só aconteceria em casos muito específicos como, por exemplo, um programa que gera outro programa que é uma versão modificada de si mesmo.

A próxima seção trata da cópia e distribuição do código fonte do Programa, que pode ser realizada desde que se mantenha os avisos sobre o *copyright*, a ausência de garantias e a licença. Nesta seção também é explicado que é permitido exigir pagamento pelo ato de transferir uma cópia ou por garantias adicionais que a pessoa decida oferecer, o que permite o uso do software em um modelo de negócio comercial.

A Seção 2 da GPL é uma das mais importantes da licença, pois é onde estão definidas as regras relacionadas ao conceito de *copyleft*, tratando das modificações do programa e sua distribuição. As condições impostas, além daquelas que foram estipuladas para a distribuição segundo o parágrafo anterior, são as seguintes:

1. “Os arquivos modificados precisam conter avisos proeminentes afirmando que os arquivos foram modificados e a data da modificação”, de forma a proteger a reputação do autor do trabalho original, que não fica sujeito a problemas causados pela versão modificada.
2. “Qualquer trabalho distribuído ou publicado que contém totalmente ou em parte o programa ou que é derivado do mesmo ou parte dele precisa ser licenciado como um todo sem nenhuma cobrança sob os termos desta licença”, o que significa que qualquer trabalho derivado está sujeito às mesmas restrições da GPL. Ou seja, não é possível que um trabalho derivado de um programa GPL seja licenciado de modo fechado, permissivo, ou qualquer outra licença, a não ser que haja a opção de utilizar outra licença para esse programa.
3. “Se o programa modificado normalmente lê comandos de forma interativa quando é

executado, é necessário que, quando ele começar a ser executado normalmente, ele imprima ou mostre um aviso incluindo as informações sobre *copyright* e ausência de garantias (ou que uma garantia é provida pela pessoa) e que os usuários podem redistribuir o programa sob essas condições e informando o usuário como ver uma cópia dessa licença”. É feita uma exceção: tal regra não precisa ser seguida no caso do programa original ser interativo mas não mostrar tal aviso. O objetivo deste item é garantir que as pessoas sejam informadas sobre seus direitos, de forma que possam utilizá-los da forma como lhes for mais interessante.

Em seguida, a Seção 2 define, com maiores detalhes, as condições em que são aplicadas as regras acima. Porém, as explicações dadas ainda estão longe de esclarecer de fato o impacto da reciprocidade em todos os casos de combinação de um programa GPL com o de outra licença. É explicado que as condições acima aplicam-se ao trabalho modificado como um todo. Os termos da GPL não precisam ser aplicados a seções que possam ser consideradas independentes e não derivadas do programa, desde que tais seções sejam distribuídas como trabalhos separados. Por outro lado, quando essas mesmas seções são distribuídas como parte de um todo que é um trabalho baseado no programa, então a distribuição desse todo precisa estar nos termos da GPL, cujas permissões se estendem para cada uma de suas partes independentemente de quem a escreveu. Isso implica que, para que o trabalho possa ser distribuído, cada uma de suas partes precisa no mínimo ter uma licença que seja compatível com os termos da GPL, já que no trabalho como um todo tais condições serão aplicadas.

A Seção 10 da GPL, que será vista adiante, provê algumas instruções sobre o que fazer quando as licenças não são compatíveis. Para evitar tal problema, alguns projetos optam por distribuir as partes necessárias para seu funcionamento separadamente, instruindo os usuários sobre como obter as partes que não estão sob seu controle e que poderiam gerar problemas de licença. É explicado que a intenção da Seção 2 não é reivindicar direitos de trabalhos criados inteiramente por outra pessoa, mas sim exercer o direito de controlar a distribuição de trabalhos derivados ou coletivos baseados no Programa. A seção finaliza afirmando que a mera agregação do Programa com outro trabalho que não seja baseado no Programa em um volume de armazenamento ou mídia de distribuição não faz com que o outro trabalho caia no escopo desta licença (GNU General Public License, version 2, 1991). Isso evita interpretações que exagerem nas consequências da licença, limitando seu alcance a trabalhos que realmente são de alguma forma derivados do programa licenciado sob a GPL.

Na próxima seção, é determinado que para distribuir o Programa é necessário que ele esteja acompanhado do código fonte. Como alternativa a distribuir o código junto com o Programa, é permitido que seja feita uma oferta por escrito, com validade mínima de três anos, de que o código seja enviado mediante uma cobrança não superior ao custo de fazer tal distribuição física. Uma pessoa que recebeu tal oferta pode reutilizá-la em sua própria redistribuição do código, desde que seja para fins não-comerciais. Em geral, a não ser que o código seja muito grande, quem distribui programas GPL já disponibiliza o código junto com o binário, de forma a evitar trabalho e custos adicionais tanto para quem está distribuindo como para quem está recebendo. Essa seção define, ainda, que o código fonte inclui todos os módulos que ele contém, arquivos de definição de interface associados e *scripts* usados para controlar a compilação e instalação do executável.

Porém, não é necessário incluir partes que normalmente são distribuídas junto com o sistema operacional em que o executável será usado, como por exemplo bibliotecas que já são parte da linguagem de programação utilizada. Essa seção finaliza explicando que se a distribuição do executável é realizada oferecendo acesso para que ele seja copiado de um determinado lugar, então disponibilizar o código fonte no mesmo lugar conta como distribuição do código fonte.

As Seções 4 a 6 da GPL continuam a explicação sobre o funcionamento da licença. Na seção 4 observa-se que, diferentemente de outras licenças que baseiam-se principalmente nas leis de contrato, na GPL há um foco também nas leis de direitos autorais. É ressaltado o fato de que somente a licença permite que uma pessoa modifique ou distribua o Programa ou um derivado dele, já que tais atos são protegidos pelas leis de direito autoral. Se a pessoa não seguir os termos da licença, ela perde esses direitos. Porém, isso não interfere nas pessoas que receberam os direitos a partir daquela que infringiu a licença. Essas pessoas continuam usufruindo dos direitos desde que elas próprias não cometam nenhuma infração. Isso acontece porque, conforme explicado na Seção 6, quando o Programa é redistribuído, o recipiente automaticamente recebe uma licença do detentor original dos direitos, e não do seu intermediário. Essa relação criada diretamente entre quem emite a licença e quem a recebe permite também que o detentor dos direitos entre com um processo contra qualquer infrator da licença. A Seção 6 também afirma que “Você não poderá impor aos recebedores qualquer outra restrição ao exercício dos direitos então adquiridos”. Na prática, essa limitação causa a incompatibilidade de outras licenças com a GPL. Devido a isso e às regras impostas na Seção 2, software distribuído sob licenças que têm alguma restrição não pode ser combinado com software GPL.

As Seções 7 e 8 da GPL continuam afirmando a necessidade de seguir os termos da licença, mesmo na presença de outros fatores geradores de outras obrigações, como, por exemplo, decisões judiciais e leis locais. Caso não seja possível cumprir tais obrigações e ao mesmo tempo seguir os termos da licença, então não é permitido que o Programa seja distribuído. Essa parte da licença é de particular importância em países onde há patentes de software, pois no caso de uma decisão judicial tentar incluir restrições relacionadas a um software, ele não pode mais ser distribuído sob a GPL. Dessa forma, o modelo de software livre proposto pela GPL é garantido mesmo na presença de decisões judiciais que iriam contra seus princípios. É permitido ao detentor dos direitos autorais limitar a distribuição do seu Programa onde as leis locais apresentam conflitos com a GPL.

Pra finalizar, as duas seções seguintes da GPL provêem mais duas explicações sobre a licença. A Seção 9 trata da política de versões das licenças publicadas pela *Free Software Foundation* e as possibilidades de escolha da versão da licença conforme o modo como a licença é especificada no Programa. Já a Seção 10 recomenda que, se uma pessoa quer incorporar o Programa em algum outro software livre cujas condições de distribuição sejam diferentes, então ela deve entrar em contato com o autor para tentar conseguir uma permissão específica. A licença termina de forma similar às licenças vistas anteriormente, com duas seções declarando a ausência de garantias. Após os termos e condições, a licença inclui, ainda, alguns parágrafos sobre como aplicá-la ao programa a ser licenciado.

A *Free Software Foundation* informa que bibliotecas que poderão ser incorporadas em software fechado utilize a licença LGPL, que será vista adiante. Segundo ela, a GPL não permite que um programa coberto por essa licença seja ligado a software licenciado sob outros termos, conforme explicado na Seção 2. Tal interpretação é causa de controvérsia

entre os estudiosos de licenças de software livre. Lawrence Rosen, em seu livro *Open Source Licensing*, uma das principais fontes sobre o assunto, argumenta que o uso de uma biblioteca cria um trabalho coletivo e não um trabalho derivado. Portanto, não seria necessário licenciar o software como GPL quando distribuído separadamente. Como essa interpretação é contrária à da *Free Software Foundation*, autora da GPL, Rosen afirma ainda que o que importa é o entendimento entre o detentor dos direitos e a pessoa que recebe a licença. Ele cita como exemplo o caso do núcleo do Linux, em que seu autor, Linus Torvalds, apesar de usar a GPL, adotou a política de que software que é apenas combinado com o Linux não está sujeito aos termos da GPL, independentemente de como esse software é ligado ou distribuído [Ros05]. Dessa forma é possível a existência de *drivers* fechados no Linux. Porém, considerando a legislação brasileira, essa interpretação de trabalho coletivo não é cabível.

4.2.2 Vantagens e Desvantagens

A GPL é bastante conhecida, sendo a licença mais utilizada em projetos de software livre. Porém, é considerada uma licença de alta complexidade. Apesar da intenção da licença estar clara em seu preâmbulo, há vários detalhes presentes em seus termos que dificultam sua interpretação em casos específicos.

A GPL é recomendada para projetos que buscam seu crescimento através de contribuições de terceiros, dado que melhorias feitas ao software devem manter-se livres para poderem ser distribuídas. A GPL também é usada frequentemente em um modelo comercial de licenciamento duplo. Neste caso, a empresa provê o software sob a licença GPL, obtendo os benefícios relacionados ao software livre, mas ao mesmo tempo disponibiliza o software sob alguma outra licença que não imponha as restrições presentes na GPL. Dessa forma, empresas que têm interesse em usar o software de forma fechada podem obter uma licença alternativa, normalmente pagando um determinado valor para a empresa detentora dos direitos sobre o software.

Por fim, antes de adotar a GPL é muito importante verificar sua compatibilidade com licenças de outros programas que serão utilizados no projeto, de forma a evitar ter que reescrever partes do software que já estariam disponíveis sob alguma outra licença.

4.2.3 GPLv3

A mais nova versão da GPL (www.gnu.org/licenses/gpl-3.0-standalone.html), lançada em 29 de junho de 2007, após um longo período de discussão e revisão pública, foi criada para evitar algumas situações consideradas indesejáveis pela *Free Software Foundation*. Além disso, algumas partes foram reescritas de forma a adaptar a licença a novas formas de compartilhamento de programa e a deixá-la mais adequada para legislações em que os termos originais poderiam ser interpretados de forma diferente da esperada pela *Free Software Foundation*. Também foram realizadas alterações para facilitar a compatibilidade com outras licenças, em particular a Apache 2.0. A seguir serão discutidas as principais mudanças.

Um dos principais motivos para mudar para a GPLv3, segundo seus criadores, é evitar o fenômeno conhecido como *tivoização* (em referência ao aparelho TiVo, que funciona como um gravador digital de vídeos). O TiVo inclui software derivado do Linux, licenciado sob a GPL 2.0. O código está disponível e pode ser modificado, porém, tais

modificações não podem ser utilizadas no aparelho TiVo, pois ele faz uma checagem da assinatura digital do software e executa apenas as versões permitidas pelo fabricante. Essa questão de assinaturas digitais foi bastante controversa durante a elaboração da GPLv3, pois ao mesmo tempo em que as assinaturas restringem a liberdade dos usuários, defendida pela *Free Software Foundation*, elas são uma ferramenta importante para implementar segurança em alguns sistemas. A estratégia para impedir a *tivoização* é exigir que o fabricante provenha toda informação necessária para instalar versões modificadas do software no aparelho. Essa informação pode ir desde instruções básicas até chaves de autorização que possam ser necessárias. Porém, tal exigência é limitada a aparelhos considerados “produtos de usuário”. Assim, alguns equipamentos de uso específico, como por exemplo máquinas de votação, estariam isentos da necessidade de reduzir seu nível de segurança para se adequar a licença. A definição de “produtos de usuário” está presente na Seção 6 da licença, sobre distribuição na forma binária:

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

Outra mudança na GPLv3 é relacionada a mecanismos de DRM, ou *Digital Rights Management* (www.defectivebydesign.org). É sabido que a *Free Software Foundation* é contra o uso de DRM, porém, não quiseram impedir que software livre fosse utilizado para implementá-lo, já que isso estaria limitando a liberdade dos usuários. Como alternativa, decidiram atacar o tratado de *copyright* da *World Intellectual Property Organization* (WIPO), adotado em 20 de dezembro de 1996. Nesse tratado, o artigo 11, sobre obrigações a respeito de medidas tecnológicas, afirma que os países devem adotar medidas legais para reprimir tentativas de burlar sistemas tecnológicos de proteção usados por autores para restringir atos que não são autorizados, em conexão com o exercício de seus direitos de acordo com esse tratado ou a Convenção de Berna. A solução presente na GPLv3 é afirmar que qualquer trabalho sob a GPL não pode ser considerado uma “medida tecnológica efetiva”. Ou seja, é permitido incluir sistemas de DRM no software tanto quanto é permitido quebrá-lo. Tal cláusula está na seção três, citada abaixo:

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

Outro ponto tratado em maior profundidade na nova versão da GPL é a questão das patentes. Uma das motivações para as mudanças que foram implementadas na licença foi um acordo entre a Microsoft e a Novell, decorrente de uma alegação da Microsoft de que a Novell estaria infringindo suas patentes na distribuição SUSE Linux. Segundo o acordo, a Microsoft não processaria usuários por infração de patentes desde que o software fosse obtido de alguém que estivesse pagando à Microsoft para obter tais direitos. A GPLv3 é bem específica quanto ao caso e, na Seção 11, sobre patentes, afirma que uma organização não poderá distribuir trabalhos cobertos por essa licença caso faça parte de um desses acordos discriminatórios, conforme detalhado a seguir:

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Há ainda outras definições sobre patentes nessa seção, porém, elas não serão detalhadas aqui, já que não são relevantes dentro da legislação brasileira. Mas essa questão das patentes é uma das principais causas de receio por parte das empresas que detêm propriedade intelectual nessa forma. Como a comunidade de software livre muitas vezes busca o apoio dessas empresas, esse tornou-se um forte fator para frear a adoção da GPLv3.

Quanto à compatibilidade com outras licenças, a mais importante delas sendo a Apache, foram adicionados alguns termos na Seção 7 da GPL, intitulada "Termos Adicionais", de forma a permitir que software cuja licença possua certas restrições ainda assim possa ser relicenciado sob a GPL, e dessa forma ser distribuído como parte de um trabalho derivado ou coletivo:

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

* b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

* c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

* d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

* e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

* f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

Para a distribuição de código binário, foram adicionadas novas possibilidades de disponibilização do código fonte correspondente, a mais importante delas tratando do compartilhamento *peer-to-peer*. Nesse caso, basta que seja informado onde o código fonte está sendo oferecido ao público. Além disso, segundo a Seção 9, a mera transmissão do programa através de *peer-to-peer* não obriga que o usuário aceite a licença. Ou seja, apenas a pessoa que inicia o processo de distribuição *peer-to-peer* precisa se preocupar em informar sobre o código fonte, as demais estando isentas de qualquer obrigação.

Por fim, vale comentar que há algumas mudanças sutis na forma como a licença foi escrita, visando evitar ambigüidade de seus termos. Por exemplo, a palavra *distribute* foi substituída por *convey*, termo que foi definido no início da licença como “qualquer forma de propagação que permite a outras partes fazer ou receber cópias”.

4.2.4 Vantagens e Desvantagens

Considerando que essa é apenas uma atualização da GPL, as vantagens e desvantagens analisadas aqui serão em relação a sua versão anterior, que já foi discutida. A GPLv3 foi lançada para corrigir algumas brechas que foram verificados no decorrer dos anos com o uso da GPL 2.0, deixando a licença mais alinhada à visão de *copyleft* da *Free Software Foundation*.

Seu texto passou por um longo período de revisão aberta pela comunidade, em que foi possível sugerir e incorporar diversas melhorias. Sendo assim, é um texto bastante sólido e bem escrito, que está consistente com uma ampla gama de legislações, inclusive a brasileira. A revisão da GPL também serviu para alinhá-la às atuais práticas de distribuição de software, incluindo o compartilhamento em redes *peer-to-peer*.

Uma das principais vantagens da nova versão da GPL é sua compatibilidade com um maior número de licenças de software livre, principalmente a Apache, que é uma das licenças permissivas mais usadas pela comunidade. Porém, a GPLv3 também apresenta uma forte desvantagem em relação a compatibilidade: software que está sob a licença

GPL 2.0 (sem a cláusula “ou posterior”) não pode ser integrado a software GPLv3, pois essas licenças são incompatíveis.

Outra desvantagem é que várias empresas têm um certo receio em adotar a GPLv3, pois ela é uma licença relativamente nova e de alta complexidade. A cláusula que gera maior preocupação nos advogados corporativos é a relacionada às patentes.

Portanto, se o objetivo é garantir as liberdades propostas pelo movimento de software livre, a GPLv3 é mais adequada do que sua anterior. Porém, se uma maior difusão do software é o mais importante, tornando-o compatível com um maior número de licenças e incentivando seu uso em qualquer empresa, então deixar sob a licença “GPL versão 2 ou superior” pode ser uma melhor alternativa. Note que se o projeto envolver componentes cuja licença é compatível apenas com a GPLv3, então é necessário que a licença adotada seja a GPLv3, e não “GPL versão 2 ou superior”.

4.2.5 AGPL

A Affero Inc. (www.affero.org) é uma empresa que se define com a missão de “trazer a cultura de patrocínio para a Internet”. Ela provê um serviço de hospedagem de páginas pessoais para autores de diversos tipos e integra um sistema de pagamento seguro para que pessoas possam fazer doações. A Affero apóia o desenvolvimento de software livre e, em março de 2002, criou a primeira versão da *Affero General Public License*, ou AGPL (www.affero.org/oagpl.html). A AGPL é uma adaptação da GPL, autorizada pela *Free Software Foundation*, que inclui um termo sobre uso de um software através de uma rede. O parágrafo adicionado à seção 2, sobre modificação e cópia do programa e sua distribuição, é o seguinte:

d) If the Program as you received it is intended to interact with users through a computer network and if, in the version you received, any user interacting with the Program was given the opportunity to request transmission to that user of the Program's complete source code, you must not remove that facility from your modified version of the Program or work based on the Program, and must offer an equivalent opportunity for all users interacting with your Program through a computer network to request immediate transmission by HTTP of the complete source code of your modified version or other derivative work.

Esse parágrafo afirma, de forma resumida, que se no programa original os usuários que interagiam com o programa tinham a opção de pedir o código fonte completo, tal opção tem que ser mantida em qualquer versão modificada. Dessa forma, mesmo não havendo a distribuição de um binário, um aplicativo web público sob esta licença precisa se manter aberto para qualquer usuário que interaja com ele.

Em 19 de novembro de 2007, a *Free Software Foundation* lançou a *GNU Affero General Public License*, conhecida como AGPLv3 (www.fsf.org/licensing/licenses/agpl-3.0.html), uma licença diferente daquela escrita pela Affero e compatível com a GPLv3. Para permitir que um programa licenciado sob a antiga licença Affero que utilizasse a cláusula “ou posterior” pudesse ser convertido para essa nova licença, a Affero Inc. criou uma versão intermediária da licença, que determina apenas o seguinte:

This is version 2 of the Affero General Public License. It gives each licensee permission to distribute the Program or a work based on the Program (as defined in version 1 of the Affero GPL) under the GNU Affero General Public License, version 3 or any later version.

If the Program was licensed under version 1 of the Affero GPL "or any later version", no additional obligations are imposed on any author or copyright holder of the Program as a result of a licensee's choice to follow this version 2 of the Affero GPL.

As diferenças entre a GPLv3 e a AGPLv3 residem no preâmbulo e na seção 13 da licença. Enquanto na GPLv3 a seção 13 informa apenas que trabalhos que usam essas duas licenças podem ser combinados, valendo a AGPLv3 como licença do trabalho resultante, a seção 13 da AGPLv3 afirma ainda que:

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Essa nova versão da redação da cláusula visa tornar a Affero GPL ainda mais abrangente. Enquanto originalmente era colocado como critério que o programa fosse feito para interagir com usuários através de uma rede, nessa nova versão, qualquer tipo de software que interage através da rede está coberto, como por exemplo servidores de jogos, que interagem com o usuário através de um outro programa intermediário.

4.2.6 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, 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.

4.3 Recíprocas Parciais

Licenças recíprocas parciais, também chamadas de *copyleft fraco*, determinam que modificações do trabalho coberto por elas devem ser disponibilizadas sob a mesma licença. Porém, quando o trabalho é utilizado apenas como um componente de outro projeto, esse projeto não precisa estar sob a mesma licença. Alguns autores, como Simon Phipps [Sun06], utilizam a denominação licença baseada em arquivo para essa categoria, enquanto as recíprocas totais seriam licenças baseadas em projeto.

Considera-se que essas licenças são as que melhor equilibram dois importantes fatores do modelo de software livre: atração de interesse para a comunidade e força e longevidade

do código fonte disponível. Ao mesmo tempo que essas licenças permitem que os desenvolvedores utilizem o trabalho para criar software que será licenciado como preferirem, modificações e melhorias feitas ao próprio trabalho são obrigatoriamente disponibilizadas à comunidade [Sun06].

A *Free Software Foundation* recomenda o uso desse tipo de licença apenas em casos específicos. Seu argumento é a necessidade de fortalecer o software livre em detrimento do software fechado. Assim, quando as funcionalidades de uma biblioteca não estão facilmente disponíveis para uso em software fechado, seria melhor mantê-las dessa forma, utilizando uma licença recíproca total. Dessa forma, o software livre teria uma vantagem sobre concorrentes fechados. Porém, se as funcionalidades já estão ao alcance de software fechado, e portanto essa vantagem não está em questão, então uma licença recíproca parcial é recomendada, pois esse modelo ajuda a aumentar o número de usuários da biblioteca [Fre08].

Alguns advogados, como Lawrence Rosen [Ros05], defendem que o uso de bibliotecas que são apenas ligadas a um novo software não caracterizaria uma trabalho derivado, mas sim um trabalho coletivo. Ele faz uma analogia a páginas na web, em que cada uma é um trabalho com direito autoral individual, apesar de muitas vezes estarem presentes ligações de uma para a outra. Segundo ele, esse tipo de relação consiste em um trabalho coletivo. Portanto, nesse cenário, mesmo um software sob uma licença recíproca total poderia ser usado como biblioteca de outro que estaria sob outra licença. Porém, no caso brasileiro, a Lei de Direito Autoral é mais específica e impõe maiores limitações. Dessa forma, o uso de licenças recíprocas parciais faz-se necessário em casos nos quais o autor quer garantir que o desenvolvimento da biblioteca seja feito no modelo de software livre mas ao mesmo tempo quer permitir seu uso em projetos que utilizam outras licenças.

4.3.1 A Licença LGPL

A *GNU Lesser General Public License*, ou LGPL (www.fsf.org/licensing/licenses/lgpl.html), originalmente denominada *GNU Library General Public License*, foi escrita em 1991 pela *Free Software Foundation*. Assim como a GPL e a AGPL, passou por grandes modificações no final de 2007 para adequar-se à versão 3 das licenças.

Em versões anteriores, a LGPL era uma cópia da GPL com algumas modificações relativas a bibliotecas, definidas na licença como “uma coleção de funções de software e/ou dados preparada para ser convenientemente ligada com programas aplicativos (que usam algumas dessas funções e dados) para formar executáveis.” Veremos a seguir os termos da versão 2.1 dessa licença. Nessa parte do trabalho, adotaremos a versão 2.0 da GPL quando esta for mencionada.

A primeira limitação presente na LGPL que não constava na GPL é que trabalhos modificados da biblioteca precisam ser também bibliotecas, segundo o artigo *a* da seção 2. Já o artigo *d* da mesma seção busca maximizar a utilidade da biblioteca, desvinculando-a de aplicações específicas. Ele determina que:

If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

Para permitir a distribuição de bibliotecas LGPL junto com software GPL, há também uma cláusula na LGPL que afirma que pode-se optar por aplicar os termos da GPL ao invés dessa licença (a LGPL) para uma determinada cópia da biblioteca. Porém, é dito também que tal mudança é irreversível para aquela cópia, e assim a GPL é aplicada a todas as cópias subsequentes e trabalhos derivados feitos a partir dela. Além disso, caso seja feito um trabalho derivado da biblioteca que não resulta em uma biblioteca, relenciá-lo para GPL é a única alternativa.

A Seção 5 da LGPL afirma que trabalhos que apenas usam a biblioteca, considerados isoladamente, não estão sujeitos aos termos da licença. Porém, também descreve em detalhes vários casos de uso da biblioteca em que é necessário estar atento às condições da licença. A Seção 5 está copiada na íntegra a seguir:

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

Apesar dessa tentativa de especificar o que seria um “trabalho que usa a biblioteca”, em que não há restrições quanto ao licenciamento, em oposição a um “trabalho baseado na biblioteca”, que estaria sujeito aos termos da LGPL, há muitos casos em que tal diferenciação não está clara. Além dos casos omissos, é possível mais de uma interpretação para alguns casos abordados nessa seção.

Há ainda a Seção 6, que faz uma exceção às anteriores e provê termos adicionais para trabalhos que incluem partes da biblioteca na versão que será distribuída. Essa seção é em parte equivalente à Seção 3 da GPL, e sua leitura é recomendada. Segundo os termos dessa seção, se a pessoa que está distribuindo um programa que usa a biblioteca não tem permissão para distribuir algum dos componentes necessários para execução do programa e esse componente não é parte integrante do sistema operacional, então a distribuição do programa na forma executável não é permitida nos termos da LGPL.

Na próxima seção, é descrito o caso de juntar uma biblioteca coberta pela LGPL com alguma outra biblioteca, sob outra licença, para formar uma única biblioteca. As regras impostas para tanto são as seguintes: em primeiro lugar, é necessário que os termos das licenças permitam que as bibliotecas sejam distribuídas como trabalhos separados. Além disso, a parte da biblioteca que está sob LGPL deve estar disponível separadamente sob os termos da LGPL, ou sendo distribuída junto com o conjunto, ou sendo colocado um aviso informando onde obtê-la.

As demais seções da LGPL correspondem às seções 4 a 12 da GPL, apenas com as alterações necessárias para tratar de uma “Biblioteca” ao invés de um “Programa”, conforme definido nas respectivas licenças.

A nova versão da licença, LGPLv3, apresenta-se como um conjunto de termos adicionais à GPLv3, já discutida na Seção 4.2.3. Isso implica que a licença LGPL foi totalmente reescrita em relação à versão 2.1, porém seus princípios foram mantidos. As diferenças entre a LGPL 2.1 e a GPL 2.0 correspondem, em grande parte, do ponto de vista semântico, às permissões adicionais que constituem a LGPLv3.

Há seis cláusulas na LGPLv3. A primeira constitui as “definições adicionais”, que explica o que será entendido por “biblioteca”, “aplicação”, “trabalho combinado”, “código fonte correspondente mínimo” e “código da aplicação correspondente”.

Em seguida, é explicado como e em que circunstâncias uma pessoa pode distribuir um trabalho que está sob a LGPLv3 sem estar sujeito à seção 3 da GPLv3 (vide termos 1, 2, 3 e 4 da licença, que não serão citados aqui por se tratar de um texto muito extenso, porém cujos detalhes são de suma importância para o uso correto da LGPLv3):

Também são definidos os termos para criar um trabalho que constituído por bibliotecas combinadas:

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

E o último termo da LGPLv3 trate de versões revisadas da licença, informando que a *Free Software Foundation* poderá publicar versões revisadas e/ou novas da LGPL e explicando que se a pessoa recebeu o trabalho licenciado sob uma certa versão da licença “ou qualquer versão posterior”, então ela poderá escolher quais das versões da licença ela irá seguir. Se não é especificada uma versão, então a pessoa pode escolher qualquer versão já publicada.

4.3.2 Vantagens e Desvantagens

A LGPL é uma licença de alta complexidade, que requer uma observação bastante atenta dos seus termos para evitar seu descumprimento, que pode acarretar em uma ação judicial. Contextos de uso da biblioteca diferentes em geral requerem ações diferentes por parte da pessoa usando a biblioteca. Apesar de todos os detalhes presentes na licença, há ainda muita margem para interpretação de casos que não estão bem definidos. Além disso, o próprio relicenciamento de trabalhos derivados como LGPL é limitado, às vezes forçando o uso da GPL, como foi visto anteriormente.

Apesar da dificuldade em usar a LGPL, ela é uma licença amplamente adotada, pois combina características permissivas e recíprocas de forma balanceada, trazendo vantagens de ambos os modelos, conforme foi discutido na introdução desta Seção.

A escolha entre a versão 2.0 ou a LGPLv3 pode ser tomada com base nos mesmos argumentos apresentados na Seção 4.2.3, sobre a GPLv3

4.3.3 A Licença Mozilla

A *Mozilla Public License*, ou MPL, foi escrita por uma das executivas da Netscape, Mitchell Baker, que tornou-se uma das principais responsáveis pelo projeto Mozilla, atuando como CEO da Fundação Mozilla por um longo período.

A licença Mozilla é considerada bem escrita e serviu como modelo para muitas das licenças de software livre comerciais que a seguiram [Ros05]. Ela une características de licenças recíprocas e de licenças permissivas, e portanto também é categorizada como uma licença recíproca parcial. Na licença Mozilla, a delimitação é bastante clara: o código coberto pela licença deve ser redistribuído pelos termos da licença Mozilla, porém esse código também pode ser utilizado em trabalhos ampliados, que podem estar sob outra licença.

A primeira seção da licença, como em um contrato convencional, trata das definições, que são bastante precisas. A lista completa de definições pode ser vista abaixo, porém algumas merecem atenção especial:

- “uso comercial” significa qualquer distribuição ou outra forma de deixar o software disponível, não se limitando ao uso por empresas;

- “contribuidor” recebe uma definição especial nessa licença, diferindo-o tanto do desenvolvedor inicial como também dos usuários comuns que estão usando o projeto;
- “executável” é definido de forma ampla, como qualquer coisa que não é o código fonte;
- “código fonte” é definido em mais detalhes do que encontramos nas licenças vistas anteriormente. São permitidos *patches* e também comprimir o arquivo, desde que o software para descompressão esteja largamente disponível gratuitamente.

1. Definitions

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

A Seção 2 da MPL trata do licenciamento do código fonte, afirmando que o desenvolvedor inicial está concedendo uma licença não exclusiva, livre de *royalties* e válida em todo mundo. Essa licença libera a propriedade intelectual (exceto patentes ou marcas registradas) para o uso, modificação, reprodução, exibição, performance, sublicenciamento e distribuição do código original (ou porções dele) com ou sem modificações e como parte ou não de um trabalho ampliado. Para o caso das patentes também são concedidas algumas permissões, porém, diferentemente da LGPL, aplicam-se apenas ao código original e não a modificações feitas nele. Nessa seção também constam as permissões concedidas por pessoas que contribuem modificações ao código, que são bastante similares às vistas anteriormente, concedidas pelo desenvolvedor inicial.

A Seção 3 trata das obrigações no ato da distribuição, que são similares às da GPL. As principais exigências são que:

- o código coberto seja distribuído sob os termos da MPL,
- o código fonte esteja disponível,
- as modificações estejam explícitas,

- o aviso legal esteja presente no código fonte,
- na distribuição de formas executáveis, as condições anteriores sejam cumpridas para o código fonte correspondente.

A principal diferença em relação à GPL está na Seção 3.7, referente a trabalhos ampliados, que são criados combinando código coberto com outro código que não está sob a licença MPL em um único produto. Nesse caso, os requerimentos da MPL recaem apenas ao código coberto, e não ao trabalho como um todo, como seria na GPL.

A Seção 4 da MPL apresenta mais uma diferença em relação à GPL. Na impossibilidade de cumprir algum dos termos da licença por questões judiciais, basta que a situação seja explicada em um arquivo chamado LEGAL e que os demais termos sejam respeitados. Assim, o uso do código não é impedido, como aconteceria com a GPL. Além disso, na Seção 8, é dado um prazo de 30 dias para que seja corrigido qualquer descumprimento dos termos antes que a licença seja terminada.

A Seção 6 afirma que a Netscape tem o direito de alterar a licença a qualquer momento, e que a pessoa que está exercendo os direitos garantidos pela licença pode escolher se seguirá os termos em que o trabalho foi licenciado ou a nova versão. Isso significa que, por outro lado, o desenvolvedor é obrigado a aceitar as modificações de licença como termos válidos para seu projeto que foi licenciado anteriormente. A Netscape transferiu os direitos de alterar a licença para a Fundação Mozilla em 2003, apesar do texto da licença ainda não ter sido modificado para refletir essa alteração. A Seção 6 permite ainda que seja criada uma licença derivada da MPL, evitando assim que a Mozilla tenha controle sobre os termos de licenciamento do projeto, desde que fique claro na versão modificada que ela não está associada à Mozilla ou à Netscape. Além disso, a Seção 13 explica o licenciamento múltiplo, que permite ao usuário escolher qual licença, entre as fornecidas pelo desenvolvedor, ele irá adotar.

As Seções 7, 9, 10, 11 e 12 tratam de garantias, responsabilidade e outros aspectos jurídicos relacionados. Para finalizar, a MPL apresenta o seguinte quadro, para aplicação da licença:

```

EXHIBIT A -Mozilla Public License.

‘‘The contents of this file are subject to the Mozilla Public License
Version 1.1 (the "License"); you may not use this file except in
compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis,
WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
for the specific language governing rights and limitations under the
License.

The Original Code is _____ .

The Initial Developer of the Original Code is _____ .
Portions created by _____ are Copyright (C) _____
_____ . All Rights Reserved.

```

Contributor(s): -----.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [_____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

4.3.4 Vantagens e Desvantagens

A licença Mozilla encoraja trabalhos ampliados. Até mesmo alguns tipos de projetos que seriam considerados trabalhos derivados pela lei de *copyright* podem usar outra licença. Basta que sejam seguidos os termos da MPL, que em linhas gerais requerem apenas que os arquivos que contém código do trabalho original estejam sob a licença MPL.

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 dos 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.

5 Conclusão e Trabalhos Futuros

As licenças de software livre muitas vezes são documentos de alta complexidade jurídica, sendo necessário um estudo cuidadoso de seus termos para avaliar o que é permitido ou não ser feito com um software. O presente relatório teve como objetivo auxiliar pessoas envolvidas no processo de desenvolvimento de software livre a escolher uma licença para seus projetos, assim como ajudar usuários a entender as limitações a que estão sujeitos.

A categorização das licenças em três grupos (permissivas, recíprocas totais e recíprocas parciais) é bastante útil para entender as implicações que a licença adotada tem na forma como o software poderá ser usado para criar e distribuir trabalhos derivados. Licenças permissivas, como a BSD e a Apache, são recomendadas quando o objetivo é uma maior disseminação do software, permitindo que qualquer um utilize-o como desejar. Já licenças recíprocas totais, como a GPL, visam fortalecer a comunidade de software livre, usando meios para garantir que qualquer alteração do programa que seja distribuída es-

teja disponível para uso e adaptação da comunidade. Além disso, essas licenças possuem uma característica “viral”, em que programas dependentes daqueles licenciados nesse modelo também precisam usar a mesma licença quando distribuídos em conjunto. Por fim, licenças recíprocas parciais, como a LGPL, funcionam como um meio termo dos dois modelos anteriores. Nesse caso, o software é desenvolvido usando o modelo de reciprocidade, porém, dependendo das circunstâncias, é permitida a distribuição de outros programas que o utilizam sem a necessidade de aplicar a mesma licença ao conjunto.

Os diferentes modelos de licenciamento de software livre podem gerar incompatibilidades entre componentes. Essas questões sobre compatibilidade, assim como a base jurídica do licenciamento de software no Brasil, são objeto de estudo de nosso mestrado.

Referências

- [CK08] Martin Campbell-Kelly. Historical reflections will the future of software be open source? *Communications of the ACM*, 51(10):21–23, 2008.
- [DOS99] Chris DiBona, Sam Ockman, and Mark Stone, editors. *Open Sources*. O’Reilly, Sebastopol, 1999.
- [For08] Forrester. Open source paves the way for the next generation of enterprise it, Nov 2008.
- [Fre08] Free Software Foundation. <http://www.gnu.org>, 2008.
- [Lau04] Andrew M. St. Laurent. *Understanding Open Source & Free Software Licensing*. O’Reilly, Sebastopol, 2004.
- [Ope08] Open Source Initiative. <http://www.opensource.org/>, 2008.
- [Ray01] Eric S. Raymond. *The Cathedral & The Bazaar*. O’Reilly, 2001.
- [Ros05] Lawrence Rosen. *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall, New Jersey, 2005.
- [Sou08] SourceForge. <http://sourceforge.net>, 2008.
- [Sun06] Sun Microsystems. Free and open source licensing, Dezembro 2006.
- [Tau04] Cezar Taurion. *Software Livre: Potencialidades e Modelos de Negócio*. Brasport, Rio de Janeiro, 2004.
- [Web04] Steven Weber. *The Success of Open Source*. Harvard University Press, Cambridge, 2004.