

Software Livre Direito Autoral

Felipe N. Gaia¹, Lucas C. Vella¹, Matheus A. Aguiar¹, Sérgio A. Santiago¹

¹Faculdade de Computação – Universidade Federal de Uberlândia (UFU)

Av. João Naves de Ávila, 2121 Bloco B, Campus Santa Mônica

CEP: 38400-902 – Uberlândia – Minas Gerais – Brasil

{felipegaia,lvella,pit,sas}@comp.ufu.br

Abstract. *This paper presents the concepts surrounding the development and the using of free software, focusing on its legal aspects. The practices for implementing the free software movement principles are heavily based on the copy-right law, wich is mainly used to assert the users freedoms as stated by the Free Software Foudation.*

Resumo. *Este artigo apresenta os conceitos que envolvem o desenvolvimento e a utilização de software livre, com foco em seus aspectos legais. As práticas para implementação dos ideais do movimento do software livre são fortemente embasadas na lei de direito autoral, sendo esta muitas vezes utilizada de modo a garantir as liberdades dos usuários como definidas pela Free Software Foudation.*

1. Introdução

Quando se trata do termo *software* livre, muitos mal-entendidos aparecem. A concepção errônea varia de pensar que o *software* é de graça a pensar que este está sob domínio público e não há restrições quanto a sua utilização.

1.1. O que é *software* livre?

A definição de *software* livre foi criada pela *Free Software Foundation* [FSF], sendo considerado como tal qualquer programa de computador que atende às quatro liberdades para os usuários do software:

Liberdade número 0: A liberdade para executar o programa, para qualquer propósito;

Liberdade número 1: A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. Acesso ao código-fonte é um pré-requisito para esta liberdade;

Liberdade número 2: A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo;

Liberdade número 3: A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para esta liberdade.

Segundo esta definição, programas que estão sob domínio público podem ser considerados livres, entretanto não é necessário que os programas que sejam livres estejam sob domínio público. O *software* pode estar sob proteção de direitos autorais e mesmo

assim ser considerado livre. Neste caso, sua licença de uso não deve impor restrições que são contrárias às liberdades básicas. As restrições impostas nas licenças de *software* livre são normalmente utilizadas para garantir que as liberdades básicas sejam respeitadas. A essa prática de licenciamento é dado o nome de *copyleft*.

1.2. *Software* livre não é *freeware*

No termo *free software*, devido a uma má interpretação do sentido da palavra *free*, que em inglês pode significar tanto livre quanto grátis, é feita uma associação errônea entre *softwares* que são distribuídos gratuitamente – os chamados *freewares* – e *softwares* livres. É uma confusão comum pensar que *freewares* são *softwares* livres, bem como o contrário. Na verdade, estes adjetivos não possuem relação direta entre si.

Não há nenhuma necessidade de que um *software* seja gratuito para que este seja considerado livre, exemplo disto é que várias distribuições do Linux [Linux] são comercializadas livremente. Contanto que o *software* respeite as quatro liberdades, o que não inclui ser gratuito, este é considerado *software* livre.

É muito fácil encontrar pela *Internet* *softwares* que podem ser adquiridos e usados sem nenhum custo, mas que impõem restrições quanto à sua utilização, modificação e redistribuição, tornando-os incompatíveis com os preceitos do software livre. Estes *softwares*, apesar de serem *freeware*, não são livres. São exemplos destes o AVG *Antivirus*, Opera *Software*, nVidia GeForce *Drivers* e Adobe *Flash Player*.

Na prática, é difícil encontrar um *software* livre que não seja também grátis, isto se deve ao fato de que a liberdade número 2 permite que o *software* seja redistribuído, fazendo com que qualquer um que tenha pago por uma cópia possa dá-la gratuitamente a outra pessoa.

É prática comum de empresas que fabricam *software* livre venderem serviços e suporte técnico, em lugar de vender o *software*, como é o caso da *Red Hat*. Existem também empresas que disponibilizam gratuitamente uma cópia do seu *software* sob alguma licença *copyleft*, e vendem uma versão que tem menos restrições ao ser utilizada para criar *softwares* proprietários. Este é o caso da *Torus Knot Software*, com o seu *software* *OGRE 3D*, e da *Trolltech* em sua biblioteca Qt.

2. Histórico

2.1. *Copyleft*

O conceito de *copyleft*, como licença legal, apareceu quando Richard Stallman trabalhava em um interpretador de *LISP*. Depois de ser contatado por uma empresa de *software*, Symbolics, ele concordou em liberar uma versão de domínio público do interpretador. A empresa posteriormente melhorou e estendeu o interpretador e quando Stallman pediu acesso às melhorias implementadas, a Symbolics se negou a atender o pedido. Stallman, então, iniciou um trabalho com o objetivo de erradicar este tipo de comportamento e cultura de *software* proprietário.

Richard Stallman decidiu utilizar a lei de direito autoral estadunidense para impedir este tipo de prática – que ele considerava abusiva – e criou sua própria licença de *copyright*, *Emacs General Public License*, que mais tarde se tornou a *GNU Public License* [GPL].

O termo *copyleft* foi cunhado por um amigo de Richard Stallman, Don Hopkins, que lhe escreveu uma carta onde dizia, “*Copyleft – all rights reversed*”, ou, “*Copyleft – todos os direitos revertidos*”. Este termo foi utilizado para descrever obras em que a lei de direito autoral era aplicada de modo a garantir liberdades que geralmente são suprimidas pela mesma. O nome *copyleft* é uma paródia com o nome da lei de *copyright* (direito de cópia, em uma tradução literal).

Para tornar um trabalho *copyleft* primeiramente registra-se o *copyright*. Então são adicionados os termos de distribuição, que garantem a qualquer pessoa os direitos de usar, modificar, e redistribuir o programa ou qualquer programa derivado dele se e somente se os termos de distribuição não forem modificados. Desta forma, o programa e as liberdades se tornam legalmente inseparáveis.

Uma licença *copyleft* pode ser considerada forte ou fraca. Se impõe suas “restrições” a todo tipo de trabalho derivado, ela é dita forte, caso contrário diz-se que a licença é fraca. *Copyleft* fraca permite que bibliotecas de *software* sob sua proteção sejam usadas em *softwares* que não necessariamente sejam *software* livre. Licenças *copyleft* podem ainda ser diferenciadas em totais ou parciais. Uma licença total exige que todo o trabalho (exceto a licença) possa ser modificado por outros. Enquanto a licença parcial aplica suas determinações apenas a algumas partes do trabalho.

2.2. GNU

Durante os anos 70 e antes, desde o início dos computadores, todo *software* que existia era livre. O termo *software* livre ainda não existia, mas o conceito e o comportamento sim. Se alguma pessoa ou empresa queria portar e usar um programa, os programadores não colocavam barreiras para que isso pudesse acontecer e muitas vezes auxiliavam na tarefa. A situação mudou nos anos 80, quando os primeiros computadores e sistemas operacionais já tinham se tornado obsoletos. Neste momento, praticamente todo *software* era proprietário, inclusive os sistemas operacionais.

Nesse contexto, Richard Stallman, que trabalhava no laboratório de inteligência artificial do MIT e não concordava com a cultura de *software* proprietário, se viu diante de uma difícil escolha moral; se juntar ao mundo do *software* proprietário e, de acordo com suas palavras: “Fazer dinheiro deste jeito e talvez me divertir escrevendo código. Apesar de que no fim da minha carreira, eu olharia atrás para os anos de construção de muros para dividir pessoas, e sentiria que eu passei minha vida fazendo do mundo um lugar pior.” A outra escolha era deixar a área de computação.

Stallman então procurou outra saída, e se perguntando se havia alguma maneira de tornar possível mais uma vez que a comunidade de *software* livre se estabelecesse chegou a conclusão de que a primeira coisa necessária era um sistema operacional. Se não existia um sistema operacional livre, não se podia nem mesmo iniciar o uso de um computador sem recorrer a *software* proprietário. E Stallman havia trabalhado como desenvolvedor de sistemas operacionais no MIT no ITS – *Incompatible Timesharing System* – para o computador PDP-10. Portanto, ele tinha as habilidades necessárias para empreender um projeto para construir um novo sistema operacional.

Em 27 de setembro de 1983, foi anunciado publicamente o plano para o sistema operacional GNU. A seguir está o trecho inicial do anúncio feito por Richard Stallman:

“UNIX livre!

Iniciando nesta ação de graças eu vou escrever um sistema completo compatível com o *UNIX* chamado *GNU* – *GNU* Não é *UNIX* [GNU] – e fornecê-lo gratuitamente para todos que possam utilizá-lo.”

O sistema operacional *GNU*, inicialmente, foi planejado para ser compatível com o *UNIX*, adicionando melhorias onde era necessário. Stallman usou vários *softwares* livres já existentes para construir o sistema operacional, como o *X Window System* para interface gráfica, o sistema de tipografia *T_EX* e o *microkernel* Mach. Outros componentes precisaram ser criados como compiladores (*GCC*), editores de texto (*GNU EMACS*), interpretadores (*BASH*) e bibliotecas (*glibc*).

No começo da década de 90, o sistema operacional *GNU* estava completo, a exceção do *kernel*. O projeto *GNU* trabalhava – e ainda trabalha – no desenvolvimento do *kernel* chamado Hurd para completar o sistema, quando um finlandês, Linus Torvalds, criou um *kernel* que poderia funcionar utilizando todos os módulos do sistema operacional *GNU*. Esse *kernel* é o Linux, contração de Linus e *UNIX*.

A partir daí, o sistema operacional *GNU/Linux* vem ganhando força e é atualmente o segundo mais usado no mundo. Ele é mantido e atualizado pela comunidade de *software* livre, juntamente com várias empresas, contando inclusive com várias e diferentes distribuições.

2.3. Free Software Foundation

A *Free Software Foundation* [FSF] é uma organização sem fins lucrativos fundada em 1985 por Richard Stallman. O principal objetivo da organização é apoiar o movimento de *software* livre e o projeto *GNU*. Inicialmente, a mesa diretora da *FSF* contava com Hal Abelson, professor de ciência da computação no MIT; Robert J. Chassell, tesoureiro da *FSF* até 1998, quando passou a se dedicar integralmente a dar palestras sobre *software* livre; Richard Stallman, fundador e único presidente; Gerald Sussman, professor de engenharia elétrica no MIT; e Len Tower Jr, primeiro funcionário em tempo integral da *FSF* e co-autor do *GNU C Compiler* – *GCC*. Os diretores da *Free Software Foundation* são elegidos por um conselho, que também define o número de diretores. Tradicionalmente, os candidatos a diretor fazem parte desse conselho votante.

Até meados da década de 90, a principal atividade da *FSF* foi, através do capital recolhido na maior parte de doações, empregar desenvolvedores de *software* para produzir *software* livre. Depois desse período inicial, como várias empresas e indivíduos começaram a produzir também *software* livre, a *FSF* voltou seu trabalho para assuntos legais e estruturais do movimento e a comunidade de *software* livre.

A *Free Software Foundation*, praticamente em qualquer momento de sua existência, tem contado com aproximadamente uma dúzia de funcionários, dos quais alguns são voluntários que não recebem pagamento. A maioria trabalha no “QG” da *FSF* em Boston.

Atualmente a *FSF* trabalha nos seguintes projetos; projeto *GNU*, destinado a incentivar os ideais e produzir *software* livre, como por exemplo o sistema operacional *GNU*; licenças *GNU*, como a *GNU General Public License* – *GPL* –, *GNU Lesser General Public License* – *LGPL* e a *GNU Free Documentation License* – *GFDL*; *GNU Press*, ou imprensa *GNU*, que é responsável por publicar livros de ciência da computação usando

licenças de distribuição livre; o Diretório de *Software* Livre – *Free Software Directory* –, que mantém uma lista de pacotes de *software* de livre, com o objetivo de fornecer uma ferramenta de busca para este tipo de *software*. Este projeto recebe uma pequena porção de patrocínio da *UNESCO*. Além destes projetos, a *FSF* também promove seminários e serviços de consultoria (especialmente para advogados) sobre aspectos legais do uso da *GPL*, apóia campanhas contra o que eles consideram perigos a liberdade de *software*, como por exemplo patentes de *softwares* e restrições contra cópia de conteúdo digital. A organização também premia anualmente pessoas que contribuíram para o avanço do movimento e da comunidade de *software* livre, com os prêmios, “*Award for the Advancement of Free Software*” e “*Free Software Award for Projects of Social Benefit*”.

Anteriormente, dois advogados serviam como conselho legal para a *FSF*, Eben Moglen e Dan Ravicher. Em fevereiro de 2005, Moglen fundou a *Software Freedom Law Center* para oferecer representação e serviços legais para proteger *software* livre e de código aberto. Desde então, a *Software Freedom Law Center* passou a representar legalmente a *FSF*.

A *Free Software Foundation* tem três organizações irmãs, que são; a *FSFE* – *Free Software Foundation Europe* – que toma conta de todos aspectos de *software* livre na europa e foi fundada em 2001; A *FSF* Índia, também fundada em 2001, promove o uso e desenvolvimento de *software* livre na Índia; e na América Latina, mais especificamente em Rosário, Argentina, em 2005, foi fundada a *FSFLA* – *Free Software Foundation Latin America* – para também promover o desenvolvimento e conhecimento do *software* livre nos países latino-americanos.

3. Tipos de licença

Há vários tipos de licenças de *software* livre comumente utilizadas. Basicamente, a diferença entre elas está nas regras que elas impõem nos atos de redistribuir e modificar o *software*.

3.1. Licença *BSD*

A primeira licença de *software* livre que se tornou realmente popular foi a licença *BSD*, criada na Universidade de Berkeley, que impõe pouquíssimas regras para a redistribuição e modificação do *software* regido por ela. Ela diz que os créditos dos autores originais devem ser mantidos, mas não estabelece outras limitações para o uso do código. Ao desenvolver uma versão comercial de um programa sob esta licença você não tem nenhuma obrigação de disponibilizar o código fonte ou dar qualquer tipo de satisfação. Um exemplo é o MacOS X, um sistema operacional proprietário desenvolvido com base no código do *FreeBSD*.

Você pode até mesmo utilizar pedaços do código fonte em outros programas, como fez a Microsoft ao utilizar a pilha *TCP/IP* e outros componentes do *FreeBSD* no Windows NT. A versão original da licença *BSD* tem uma falha, a “cláusula de propaganda”, ela diz que todo anúncio mencionando o *software* deveria incluir os créditos originais:

“Todos materiais de propaganda mencionando recursos ou uso deste *software* devem fazer o seguinte reconhecimento:

Este produto inclui *software* desenvolvido pela Universidade da Califórnia em Berkeley, ou por seus colaboradores.”

Então desenvolvedores não copiavam literalmente a cláusula do *BSD*. Eles a alteravam, trocando “Universidade da Califórnia” por sua própria instituição, ou por seus nomes. O resultado foi uma grande quantidade de licenças, que requeriam uma grande quantidade de sentenças diferentes.

Esse problema foi corrigido em 1999, criando a nova versão da licença *BSD*, que excluía a cláusula de propaganda.

3.2. Licença *GPL*

Criada pela *Free Software Foundation*, a *GPL* estabelece regras mais rígidas para a cópia, redistribuição e alteração do *software*. Fundamentalmente, ela quer garantir que qualquer *software* livre licenciado por ela mantenha sempre esta condição, isto é, se o *software* for alterado e redistribuído, essa alteração passa a ser *software* livre também e deve estar disponível sob os mesmos termos, fazendo da *GPL* uma licença de *copyleft*. Justamente por reger também futuras extensões do *software* licenciado por ela, a licença *GPL* é comumente adjetivada de *virótica*, pois *contamina* qualquer novo trecho de *software* que *entra em contato* com *software* regido por ela.

O rigor imposto pela licença *GPL* gerou grande polêmica na comunidade de *software* livre pois alguns desenvolvedores de *software* consideraram que ela restringia a liberdade oferecida, por exemplo, pela licença *BSD*, de se utilizar *software* livre como parte de um produto comercial. Desta forma, foi criada a *Open Source Initiative* [OSI] que, entre outras coisas, passou a validar através de um selo de aprovação as licenças consideradas livres o suficiente. O mais curioso é que a própria licença *GPL*, que causou boa parte da polêmica, também acabou recebendo mais tarde esse selo de aprovação.

3.3. Licença Apache

Licença criada pela *ASF* – *Apache Software Foundation*. Todo *software* produzido pela *ASF* ou qualquer um dos seus projetos e sub-projetos é licenciado de acordo com os termos da licença Apache. A licença Apache exige a inclusão do aviso de *copyright*, mas não é uma licença *copyleft*, ela permite o uso e distribuição do código fonte tanto no *software* livre como no proprietário.

A licença Apache tem alguns problemas de compatibilidade com a *GPL*, na última versão da licença Apache, a *Free Software Foundation* disse: “A licença Apache é incompatível com a licença *GPL* porque ela tem um requerimento específico que não está na licença *GPL*: ela tem certas condições relacionadas à patentes e que determinam o término da licença, que a *GNU GPL* não tem. Nós não consideramos essas condições como idéias ruins, apenas que elas são incompatíveis com a *GNU GPL*.”

3.4. Licença *MIT*

Foi criada pelo *Massachusetts Institute of Technology*, e recebe o nome de licença X ou de licença X11. Ela é uma licença não *copyleft* utilizada em *software* livre, isto é, ela permite a reutilização de *software* licenciado em programas livres ou proprietários. Não há *copyright* no texto da licença *MIT*, por isso, outros grupos podem modificar a licença para atender as suas necessidades.

Segundo a *Free Software Foundation* [Stallman 1999], a licença *MIT* deveria ser chamada de licença X11, já que o *MIT* possui muitas licenças para *software*. Por outro lado, a *Open Source Initiative* se refere a ela como licença *MIT*, assim como outros.

3.5. Creative Commons

A *Creative Commons* é uma organização sem fins lucrativos dedicada a expandir a quantidade de trabalho criativo, não só *softwares*, mas também de obras musicais, literárias entre outras. Ela foi lançada oficialmente em 2001 por Lawrence Lessig, o fundador e presidente da associação. As licenças disponibilizadas pela *Creative Commons* têm como principal objetivo, dar aos autores das suas obras, opções para que elas possam ser, compartilhadas, modificadas, usadas e redistribuídas desde que citada a fonte com os devidos créditos. A disponibilidade da nova criação também deve seguir atributos similares ou licença compatível.

O principal lema da *Creative Commons* é, partindo da idéia de “todos os direitos reservados” do direito autoral tradicional foi recriado para transformá-la em “alguns direitos reservados”.

Existem vários tipos de licenças disponibilizadas pela *Creative Commons*, para cada perfil de autor diferente. As principais licenças são:

3.5.1. Atribuição - Uso Não Comercial

Esta licença permite que outros *remixem*, adaptem, e criem obras derivadas sobre uma determinada obra sendo vedado o uso com fins comerciais. As novas obras devem conter menção ao autor original nos créditos e também não podem ser usadas com fins comerciais, porém as obras derivadas não precisam ser licenciadas sob os mesmos termos desta licença.

3.5.2. Atribuição - Não a Obras Derivadas

Esta licença permite a redistribuição e o uso para fins comerciais e não comerciais, contanto que a obra seja redistribuída sem modificações e completa, e que os créditos sejam atribuídos ao autor original.

3.5.3. Atribuição - Compartilhamento pela mesma Licença

Esta licença permite que outros alterem, adaptem, e criem obras derivadas ainda que para fins comerciais, contanto que o crédito seja atribuído ao autor e que essas obras sejam licenciadas sob os mesmos termos. Esta licença é geralmente comparada a licenças de *software* livre. Todas as obras derivadas devem ser licenciadas sob os mesmos termos desta. Dessa forma, as obras derivadas também poderão ser usadas para fins comerciais.

3.5.4. Atribuição

Esta licença permite que outros distribuam, alterem, adaptem ou criem obras derivadas, mesmo que para uso com fins comerciais, contanto que seja dado crédito pela criação original. Esta é a licença menos restritiva de todas as oferecidas, em termos de quais usos outras pessoas podem fazer de sua obra.

3.6. GPL versão 3

Com o passar dos anos, a evolução tecnológica mudou o modo como os softwares são distribuídos e as informações são passadas, tornando os termos e requerimentos da *GPL* versão 2 obsoletos. Também foram desenvolvidos alguns artifícios para burlar as restrições de *copyleft* da *GPL* versão 2. Para resolver estes problemas, a *Free Software Foundation*, em conjunto com a comunidade, desenvolveu a *GPL* versão 3.

Uma das técnicas para tornar um *software* sob *GPL* proprietário que foi impedida na *GPL* 3 consiste em utilizar o programa em um *hardware* que não provê meios de o usuário alterar o programa nele contido, tendo esta opção somente o fabricante do *hardware*. Este é o caso da maioria dos sistemas embarcados que utilizam *softwares* sob *GPL*.

Outro abuso que era possível com a *GPL* 2 mas não é mais possível na versão 3, é utilizar das leis de patentes de *software*, presentes em alguns países, para restringir sua liberdade. Se alguém acrescentasse alguma técnica de *software* protegida por uma patente a um *software* sob *GPL*, na versão 2, era permitido distribuir o *software* somente a quem possuía direito de utilização daquela patente. Já a versão 3 garante que uma patente não pode tornar o *software* não-livre.

Quando a *GPL* 3 estava em seus estágios iniciais de desenvolvimento, seus rascunhos foram severamente criticados pelos desenvolvedores do Linux [Bottomley et al. 2006], principalmente no que se refere a cláusula de restrições adicionais (que permite que o detentor dos direitos autorais adicione algumas restrições adicionais à licença), *Digital Rights Management* e medidas contra patentes.

Muito do que foi criticado nos rascunhos da *GPL* 3 foi abrandado ou corrigido em sua versão final, publicada dia 29 de junho de 2007 pela *Free Software Foundation*. As modificações feitas levaram Linus Torvalds, um dos maiores críticos da licença até então, a reconsiderar sua posição com relação ela [Shankland 2007].

4. Open source versus software livre

Muito se discute sobre as diferenças entre *open source* e *software* livre, porém como já foi dito a própria licença *GPL* recebeu o selo da *OSI* – (*Open Source Initiative*). Na prática os dois movimentos são a mesma coisa no que tange à questões práticas (como o modelo de desenvolvimento). As pequenas diferenças existentes entre eles estão ligadas a questões morais.

O movimento do *software* livre é um movimento social que defende que o conhecimento científico não deve ser aprisionado, ele deve estar sempre disponível, permitindo a evolução da humanidade, já o movimento *open source*, que não é um movimento social, porém voltado ao mercado, defende que o *software* produzido dessa forma possui muitas vantagens técnicas e econômicas.

5. Como tornar seu código livre?

Como obra protegida pelo direito autoral, e passível de suas implicações, é recomendável que um programa de computador seja registrado junto a um órgão governamental de modo a facilitar a comprovação da autoria da obra em um eventual processo jurídico.

No Brasil, o órgão responsável pelo registro de *software* é o INPI – Instituto Nacional de Propriedade Industrial. Entretanto, o registro não é uma obrigatoriedade: qualquer programa escrito já estará automaticamente protegido pela lei de direito autoral a não ser que o contrário seja explicitamente especificado pelo autor do programa.

Uma vez possuído o *copyright* sobre um programa, e estando este pronto para ser distribuído, o primeiro passo a ser dado consiste em colocar um comentário em cada arquivo do código fonte dizendo quem é o detentor dos direitos autorais para aquele trabalho e em que ano foi feito. Feito isso, é necessário decidir em quais termos publicar o *software* como livre.

O simples fato de colocar um comentário no topo de arquivo do código fonte, dizendo que o mesmo está sob domínio público, portanto, abrindo mão o autor dos seus direitos exclusivos sobre a obra, já define o *software* como livre. Esta não é uma prática comum, pois possibilitaria que modificações feitas no *software* em questão tivessem seus direitos restringidos pelo autor das modificações.

Pode ser usada uma licença de uso para distribuição do programa. Neste caso, deve-se decidir qual licença usar. A escolha de uma licença *copyleft* asseguraria que qualquer *software* derivado continuasse a ser livre. Nesta categoria estão incluídas a *GPL* e a *LGPL*.

Uma licença menos restritiva poderia ser usada, de modo a apenas dar algumas garantias e direitos a mais para autor do que simplesmente colocar o *software* em domínio público, mas não garantiria que o *software* seria livre em todas as suas gerações futuras. As licenças *zlib License* e *BSD* modificada são exemplos de licenças que se enquadram nesta categoria.

De modo geral, para distribuir um *software* sob uma licença específica, é necessário que em cada arquivo do código fonte seja adicionado um cabeçalho especificando os detentores dos direitos autorais, sob qual licença o *software* é distribuído e como obter uma cópia da licença. Deve ser incluída, junto com o pacote da distribuição do *software*, uma cópia da licença.

Licenças de *software* livre são normalmente feitas de um modo genérico e não necessitam ser alteradas para que possam ser utilizadas em vários programas diferentes. Sua utilização em *softwares* de terceiros é geralmente permitida e incentivada pelos autores da licença. Entretanto, como uma licença em si também é um material sob *copyright*, elas não devem ser modificadas ou utilizadas sem a permissão do autor. Nos casos específicos da *GPL* e da *LGPL*, a *Free Software Foundation* permite a livre utilização das licenças, mas não permite que estas sejam modificadas.

6. Como e onde publicar seu *software* livre?

Ao seguir os procedimentos já descritos no tópico anterior e ter tornado seu programa um *software* livre, podemos publicá-lo, ou seja, disponibilizá-lo para a comunidade. Para isso existem alguns mecanismos que auxiliam nesse processo, eles fornecem alguns serviços de um servidor de hospedagem, podemos destacar dois desses mecanismos:

SourceForge [SourceForge], criada pela VA Linux;

Savannah [Savannah], criada pela *FSF* (*habitat* natural do *GNU*).

Vamos entrar em maiores detalhes sobre o primeiro.

6.1. SourceForge

É um Provedor de Serviço de *Internet (ISP)* gratuito que hospeda projetos de *software* livre. Tem sido um dos mais utilizados para essa finalidade por possuir muitos recursos tanto de hospedagem quanto de desenvolvimento. Entre os serviços de hospedagem disponíveis podemos destacar: espaço em disco; *web site* e *FTP*; aplicativos *CGI* e bancos de dados; listas de *e-mail*; fóruns de discussão.

E entre os recursos de desenvolvimento destacamos: repositórios de código gerenciados por *CVS*; *building farms* – servidores onde os desenvolvedores podem gerar binários para diversas plataformas; *bug tracking* – sistema de registro e acompanhamento de *bugs*; sistema de alocação e acompanhamento de tarefas entre os desenvolvedores; *feature request* – onde os usuários podem requisitar e votar em novas implementações.

6.1.1. Como participar do SourceForge?

Primeiramente seu projeto deve ser liberado sob uma das licenças aprovadas pela *Open Source Initiative*. Você deve registrar você e o seu projeto em uma página *web*, cedendo ao *SourceForge* os direitos de: continuar hospedando uma versão (possivelmente obsoleta) do seu projeto caso você decida sair do *site*; continuar a desenvolver o seu projeto sob a licença previamente escolhida por você, caso você decida reverter o projeto para uma licença proprietária.

Caso seu projeto seja aprovado você receberá uma notificação avisando o confirmamento do seu cadastro e seu espaço para *web* e *FTP* estará liberado. Você poderá utilizar seu usuário criado no registro do projeto para gerenciar listas de *e-mail*, fóruns de discussão e para autorizar outros usuários cadastrados no *SourceForge* a participar do seu projeto. Você deverá utilizar *scp* para publicar o conteúdo de seu *site*, que terá como nome: `http://seu-projeto.sourceforge.net/`.

Após isso você deverá ter instalado em sua máquina o *CVS* para controlar a versão do seu código no repositório do seu projeto no *SourceForge*, assim todas as modificações feitas por você (ou por outro desenvolvedor autorizado) serão gerenciadas por *CVS*. Você pode também criar subprojetos no *CVS*. O *SourceForge* também fornece uma interface *web* para navegação no repositório e envio de notificações.

Referências

Bottomley, J. E., Chehab, M. C., Gleixner, T., Hellwig, C., Jones, D., Kroah-Hartman, G., Luck, T., Morton, A., Myklebust, T., and Woodhouse, D. (2006). *Kernel developers' position on GPLv3*. URL: `http://lwn.net/Articles/200422/`. Acesso em 31 de julho de 2007.

FSF. *Free Software Foundation*. URL: `http://www.fsf.org`.

GNU. *GNU's Not Unix*. URL: `http://www.gnu.org`.

GPL. *GNU Public License*. URL: `http://www.gnu.org/copyleft/gpl.html`.

Linux. *The Linux Kernel Archives*. URL: `http://www.kernel.org`.

OSI. *Open Source Initiative*. URL: `http://www.opensource.org`.

Savannah. URL: <http://savannah.gnu.org>.

Shankland, S. (2007). *Torvalds 'pretty pleased' about new GPL 3 draft*. URL: http://news.com.com/8301-10784_3-6171300-7.html. Acesso em 31 de julho de 2007.

SourceForge. *SourceForge.net*. URL: <http://sourceforge.net>.

Stallman, R. M. (1999). *Various Licenses and Comments about Them*. URL: <http://www.gnu.org/licenses/license-list.html>. Acesso em 31 de julho de 2007.