



CENTRO DE ENSINO UNIFICADO DE BRASÍLIA - UnICEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO
DISCIPLINA SISTEMAS OPERACIONAIS

ESTUDO JURÍDICO SOBRE AS LICENÇAS DE SOFTWARES LIVRES
(AGPL, GPLv2, GPLv3 e MIT);

Autor:
Altair Correia de Azevedo

Brasília, DF (2020)

APRESENTAÇÃO

Para que todo software (proprietário, livre, de código aberto, gratuito ou, ainda, *copyleft*) seja instalado em uma máquina, com suas devidas formalidades legais preenchidas, é necessário que ele possua licenças próprias, devidamente aceitas pelo usuário, antes mesmo da instalação, pois caso não concorde com os termos legais previstos na licença, este software, normalmente, não prossegue na sua instalação.

De fato, isso vale também para um sistema operacional (o software básico de qualquer máquina) que por mais fundamental que seja, ainda é um software, e sem ele, obviamente, um computador é tão somente um emaranhado de peças de hardware unidas sem utilidade alguma.

Dito isto, vem em mente a seguinte pergunta: o que vem a ser um software?

De acordo com a Lei n. 9.609/1998, no Brasil foram estabelecidas normas especiais para a proteção da propriedade intelectual de programa de computador e sua comercialização, além desta lei definir software como (PIMENTEL, 2014)

[...] a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital ou análoga, para fazê-los funcionar de modo e para fins determinados. (BRASIL, 1998, art. 1º)

Uma definição de software, mais técnica, segundo GUTIERREZ; ALEXANDRE(2004) é um:

Conjunto de instruções e dados que permitem a um computador a realização de tarefas previamente programadas, seja por uma máquina (hardware) ou outros softwares.

Com isso, naturalmente vem outra pergunta: o que é uma licença?

A grosso modo, é tão somente um texto legal detalhado que informa e delimita quais direitos e deveres – de uso, de modificação, de distribuição, etc. – a pessoa que adquiriu algum software tem sobre esse mesmo software, sendo ele livre ou proprietário. Explicando de outra forma, a licença de software é “(...)o documento que define os limites de uso que um usuário pode ter em relação a um produto de terceiros. (...) compreendido pelo Direito, e pode restringir a cópia, a distribuição e a adaptação da aplicação.”

Nesse estudo, o Linux, por não ter sido desenvolvido com fins comerciais, sendo seu código fonte aberto a todos que queiram modificar ou criar novas distribuições a partir dele, é a base do estudo jurídico das licenças de software livre explanadas nesse trabalho.

As licenças seguem a diversidade dos softwares criados, mas pode-se dividir em dois grandes tipos: (1) licenças de software proprietário e (2) licenças de software livres. Como dito, existe uma diversidade de licenças de softwares disponibilizadas no mercado, mas neste estudo jurídico iremos nos ater às seguintes licenças de softwares livres AGPL, GPLv2, GPLv3 e MIT em distribuições Linux como base no desenvolvimento do sistema operacional do UniCeub, denominado como CEUB-OS.

OBJETIVO

Esse trabalho foi desenvolvido com o intuito de apresentar e esclarecer para qualquer pessoa o que vem a ser uma licença de software para uso em sistemas operacionais baseados em Linux, se atentando, oportunamente, aos dizeres legais e normas de utilização do sistema operacional CEUB-OS e sua importância para a disciplina de Sistemas Operacionais, dando ênfase às licenças de softwares livres.

SOFTWARE PROPRIETÁRIO/FECHADO

Antes do estudo das licenças de software livre propriamente dito, vale realizar uma breve explanação do que vem a ser o software proprietário. Esse tipo de software é pago e muitas vezes de modificação bem restrita ou inacessível a qualquer alteração. Não é objeto desse estudo, portanto, mas é um contraponto direto às quatro “liberdades essenciais” da definição do software livre.

Dessa forma, para não deixar uma lacuna no andamento do estudo e servir de ponto de comparação, passa-se a realizar um breve histórico do que é o software proprietário, também conhecido como software fechado.

Segundo (SABINO; KON, 2009), a ideia de software fechado é advinda da década de 1970, com o surgimento de várias marcas fabricantes de software que precisavam fazer uso nos diversos hardwares que eram criados, mesmo ainda nos hardwares mais básicos que usavam programas desenvolvidos pelo próprio fabricante desse hardware.

Nesse período, o que simplesmente ocorria era a prática da “venda casada” (atitude ilegal como é hoje estabelecida pelo código do consumidor), mas necessária, pois os programas eram profundamente acoplados (embarcados) à arquitetura das máquinas em que eram executados. Basicamente, o hardware já chegava com o software que o faria funcionar já incluso no preço final do equipamento. Em outras palavras, o foco das empresas era na venda do hardware, e não eram colocadas muitas restrições no uso que as pessoas fariam do software que eles traziam.

Assim, segundo o autor em seu trabalho,

(...) algumas empresas, destacando-se 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.

Nessa carta, basicamente, 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 (vide imagem 1, abaixo).

Com isso, nessa época começou uma mudança de postura na indústria, que passou a proibir que o software fosse copiado ou modificado. Surgiu então o software caracterizado pelas restrições que são feitas à forma como ele seria utilizado e que se estende até os dias de hoje.

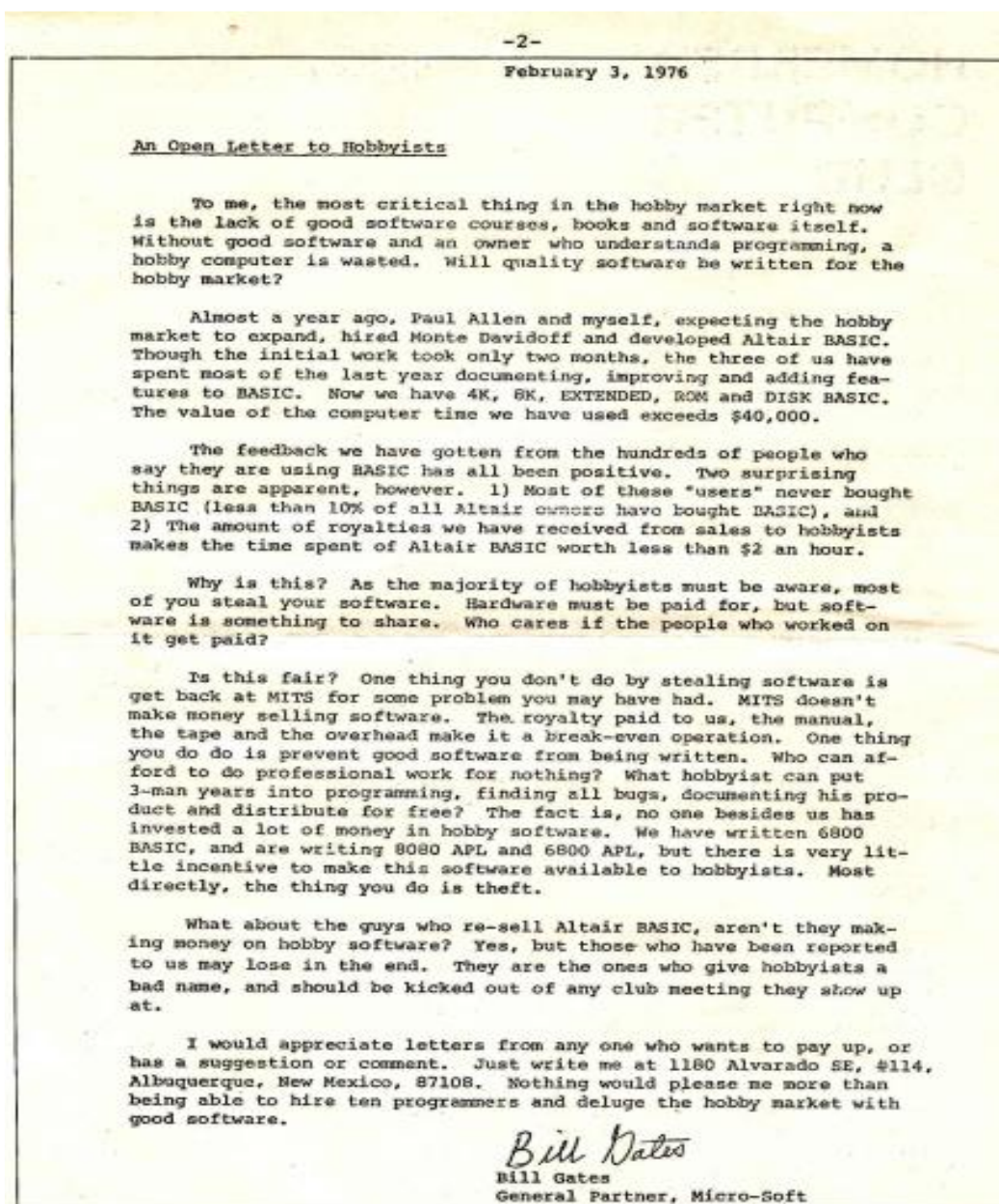


Imagem 1: Carta escrita por Bill Gates e publicada na Homebrew Computer Club

Como um revide a essa nova conjuntura, surgiram iniciativas voltadas para retomar as liberdades essenciais do software livre, além de melhorar e compartilhar o software mais massivamente. Trataremos a seguir de duas dessas iniciativas. São elas: o Projeto GNU, que apregoa a filosofia do Software Livre, e o movimento Open Source, atrelado aos softwares de código-fonte aberto.

SOFTWARE LIVRE: Projeto GNU/Linux

Richard Stallman, em 1983, informa, por meio de grupos de trabalho, nos primórdios da internet, que está começando a escrever um novo sistema operacional compatível com o UNIX, o qual chamou inicialmente de GNU e que ele será distribuído às pessoas que se interessarem. O sistema basicamente teria o núcleo próprio, um compilador C e um editor de texto, além de propor melhorias em relação ao próprio UNIX existente na época.

Ele decide criar seu sistema utilizando todo o tipo de software livre disponível para que pudesse ser criado, modificado e distribuído sem restrições com softwares considerados não livres. Assim, ele solicita ajuda e contribuições para escrever o código, máquinas e recursos financeiros para financiar o projeto, para que, no final, possa ser distribuído a quem se interessar.

Segundo (SABINO; KON, 2009), em 1984, Stallman deixou o trabalho no laboratório de Inteligência Artificial do MIT, querendo não perder os direitos sobre seu projeto, e começou a dedicar-se em tempo integral ao GNU. Para ele, esse negócio de distribuição de software livre na época, foi um precursor das empresas que hoje distribuem o GNU/Linux. Desenvolveu, com a adição de outras linguagens, o chamado GNU *Compiler Collection* que é um dos componentes mais importantes do sistema GNU atualmente. Stallman decidiu incorporar, ao sistema GNU, softwares que não foram escritos propriamente pelo projeto GNU, como por exemplo o Linux e o X Window System. Tudo foi possível, pois esses programas eram considerados livres também.

Ao mesmo tempo que o sistema GNU era formado, conceitos e especificidades dos softwares livres foram estabelecidos, o que acabou por culminar com a criação da *Free Software Foundation*, em 1985. Assim, o conceito de software livre foi definido como “o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição.”, além desse conceito, ficou estabelecido as bases ou liberdades dos softwares livres:

1. Liberdade para executar o programa, para qualquer propósito;
2. Liberdade para estudar e modificar o programa;¹
3. Liberdade para redistribuir o programa, gratuitamente ou pago; e
4. Liberdade para melhorar e redistribuir as melhorias à comunidade.

¹ Para tornar essa liberdade efetiva na prática, o desenvolvedor precisa ter acesso ao código fonte, já que fazer alterações em um programa sem ter tal código é muito difícil.

Essas liberdades precisavam ser garantidas e com esse intuito foi criado o sistema de *copy/left* para impedir que os sistemas que usem softwares livres voltassem a ter o caráter de sistema fechado/proprietário. Mais a frente, detalha-se melhor tal sistema.

SOFTWARE LIVRE: Movimento *Open Source*

Interessante, nessa época, como os usuários e desenvolvedores, com a ideia do software livre, entenderam que a característica principal do software livre, criado pela *Free Software Foundation*, era ter conotações ditas “anticapitalistas” pelo uso da palavra “livre” simplesmente por se entender que indicava liberdade e/ou libertação do capitalismo.

Com isso em mente, a ideia foi alterar a abordagem de como seria oferecido o software livre para pessoas mais conservadoras, e criou-se o termo *Open Source* em 1997. Não usando a palavra *free*, estava não apenas evitando a confusão com gratuito, como também tirando a sensação esquerdista do termo proposto por Stallman. Com o *Open Source* em ascensão, que destacava os acréscimos técnicos advindos dos métodos assimilados e distribuídos pela comunidade, e tendo como principal exemplo a *Netscape*, primeira empresa a abrir seu código-fonte à quem se interessasse, fez com que o software livre fosse usado por parte das empresas, disseminando e gerando um grande impulso.

Abaixo a definição oficial, com as suas condições, do que vem a ser o código aberto:

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.

Imagem 2: Tradução livre da *Open Source Definition* publicada em opensource.org/docs/osd)

Com base nessa definição mais detalhada, a OSI (*Open Source Initiative*) aprova as licenças que podem ser consideradas *open source*. A aprovação de uma licença está condicionada à aferição, por parte da comunidade, em processo público, dada a devida publicidade, revisando os termos da licença, conferindo se há conformidade com as cláusulas e expectativas previstas na definição *Open Source*.

LICENÇAS DE SOFTWARE

As licenças são classificadas em 3(três) grupos de acordo com a forma como podem ser em relação às suas restrições:

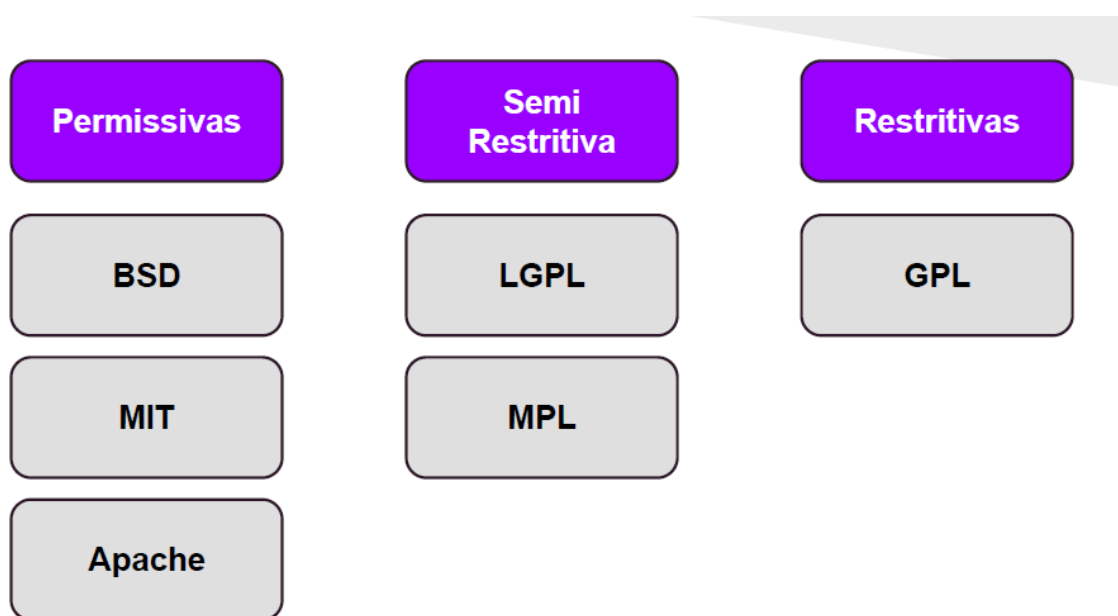


Imagem 3: Classificação das principais licenças de software existentes

As licenças permissivas também denominadas de licenças acadêmicas, por terem origens na universidade americana de Berkeley, Califórnia (licença BSD) e pelo Instituto de Tecnologia de Massachusetts (licença MIT) impõem poucas ou nenhuma restrição às pessoas que obtêm o produto. Essas licenças são basicamente buscadas pelas universidades para criação de projetos de pesquisa derivados ou mesmo de aperfeiçoamento do original, sem que se preocupem se essa venha a se tornar algo comercial no futuro. Como não há restrições, nada impede de os projetos baseados nessa licença possam se tornar produtos de licença fechada. Além das citadas licenças BSD e MIT está inclusa nesse grupo a licença Apache, dentre outras.

Licenças permissivas são muito utilizadas para projetos com o intuito de capilarizar o conhecimento ao maior número de pessoas, tanto com software livre ou por meio de software fechado. Temos como exemplo mais notório desse modelo de licença no BSD Unix, que continha as bases do protocolo TCP/IP que hoje é usado na maior parte das implementações desse protocolo na internet que conhecemos hoje. Mais um exemplo: o BIND (Berkeley Internet Name Daemon), que serviu para a implementação livre dos

principais servidores DNS (Servidores de Nome de Domínio), que transforma todos as URLs da web em formato IPV4 ou IPV6 em nomes mais simples de serem digitados para as páginas que deseja acessar.

Já as licenças semi-restritivas ou recíprocas parciais conhecidas, também, como *copyleft* fraco, determinam que modificações do trabalho coberto por elas devem ser disponibilizadas sob a mesma licença. Ou seja, caso haja modificação esta deve ser repassada para outros sob a mesma forma que foi recebida na licença original. Contudo, se o projeto utilizar um componente qualquer da licença original, sem utilizar de forma integral, esta pode ser disponibilizada sob licença diferente. Alguns autores utilizam a denominação de licença baseada em arquivo para essa categoria. São consideradas o melhor dos “mundos” do software livre, são equilibradas, na verdade, pois demandam dois importantes fatores do modelo de software livre: interesse para a comunidade e força/longevidade do projeto em si. Aqui nesse grupo inclui-se a licença LGPL (GNU Lesser General Public License), e a MPL (Mozilla Public License), dentre outras.

Por fim, temos as licenças restritivas ou recíprocas totais que determinam que qualquer projeto derivado precisa ser distribuído sob os mesmos termos da licença original. Também conhecida como *copyleft*, conceito criado pela *Free Software Foundation*. A licença que deu origem à ideia de *copyleft* foi a GNU General Public License, comumente chamada apenas de GPL, atualmente na versão 3. A ideia do *copyleft* é se contrapor ao *copyright*, ou seja, permitir a todos que execute, copie, modifique e distribua o programa, alterado ou não, mas impõe que haja restrições quando a redistribuição do novo software. A intenção com tudo isso é fazer com que o software livre, depois de melhorado, acrescido com atualizações em seu programa possa continuar sendo compartilhado continuamente. É o princípio que estabelece a melhoria contínua do software livre.

Segundo SABINO; KON(2009), complementando:

Além disso, a reciprocidade contribui para manter a compatibilidade entre diversas versões de um determinado sistema, dado que quando novas funcionalidades são introduzidas de forma restrita fica mais difícil replicá-las em outras versões e derivações. 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 por 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 disponibilizado sob licenças recíprocas não pode ser utilizado em projetos de software livre que usam alguma outra licença se esta não for reciprocamente compatível com aquela. (grifo meu)

Neste grupo, como visto, estão as licenças GPL em suas posteriores versões, a AGPL, a LGPL, dentre outras.

Iremos tratar a partir daqui, neste estudo, das seguintes licenças:

- MIT licence;
- GNU General Public License (GPL) – atualmente na versão 3;
- Affero General Public License (AGPL)

A LICENÇA MIT ou X11

Esta licença, criada pelo Massachusetts Institute of Technology, daí ser chamada de licença MIT e é também conhecida como Licença X11 ou X, por ter sido redigida para o X Window System, desenvolvido no MIT em 1987. Por ser uma licença permissiva, pode ser considerada equivalente à licença BSD, porém sem a primeira restrição prevista na letra (“a”) do texto original da BSD.

Os autores SABINO; KON(2009) explica que:

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 de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do software. As condições impostas para tanto são apenas manter o aviso de *copyright* e uma cópia da licença em todas as cópias ou porções substanciais do software.

Assim, essa licença é a recomendada pela *Free Software Foundation* para o caso de projetos que precisem de uma licença permissiva, pois é muito experimentada e, ao contrário da BSD, não possui inúmeras versões, originando dificuldades na sua confecção, com muitas cláusulas adicionais, podendo gerar até algumas incompatibilidades com outras licenças na sua distribuição.

SABINO; KON(2009) ainda explana a devida atenção com o uso dessa licença, pois

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 ideia 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.(grifo meu)

Abaixo veja na íntegra, a licença MIT², disponibilizada no site da OSI:

² Disponível em: <https://opensource.org/licenses/mit-license.php>

The MIT License

License Copyright: Unknown.
License License: Unknown.
License Contact: Unknown.

SPDX short identifier: MIT

[Further resources...](#)



Begin license text.

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

End license text.

Imagem 4: Texto original, na íntegra, dos termos da Licença MIT.

Vantagens e desvantagens

Essa licença é, como dito anteriormente, uma licença permissiva, muito conhecida e sem múltiplas versões como a BSD possui, que culmina em gerar confusão, usando muitas cláusulas adicionais, tornando-se esse fator simplificador da MIT uma imensa vantagem.

Outra vantagem da MIT em relação à BSD é a maior clareza dos seus termos ao declarar explicitamente que é lícito sublicenciar ou vender cópias do software. Na BSD essas informações aparecem apenas de forma implícita, podendo gerar interpretações ambíguas e ocasionar alguns transtornos no futuro.

Ainda sobre o sublicenciamento, SABINO; KON(2009), reforça que:

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 detentor do direito autoral pode conceder a licença.

Por outro lado, como desvantagem, a licença MIT exige do usuário de um projeto derivado a obtenção de uma licença tanto do autor desse projeto 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. Mas essa desvantagem desaparece caso seja permitido o sublicenciamento, podendo ser delineado a cada um dos componentes do software em seu texto. Dessa forma, ao chegar no usuário final, o potencial de problemas é restrito a uma única licença.

A LICENÇA GPL

A licença GPL foi escrita em 1989 pela *Free Software Foundation*. Em junho de 1991, foram feitas pequenas modificações na licença, gerando a versão 2.0 que ficou vigente até 2007, quando foi definida a GPLv3, válida até os dias atuais.

A GPL é uma das licenças mais utilizadas em projetos de software livre, pois é aconselhada para projetos que buscam o crescimento através de apoios de terceiros, dado que melhorias feitas ao software devem manter-se livres para poderem ser redistribuídas.

SABINO; KON(2009), informa que:

Segundo a GPL, a cópia e distribuição do código-fonte do programa, com ou sem modificações, pode ser realizada desde que se mantenham os avisos sobre o copyright, a ausência de garantias e a licença. Já no caso da distribuição de um binário, é obrigatório que ele acompanhe o código-fonte ou instruções de como obtê-lo ao custo máximo do meio físico utilizado para transferi-lo. A licença explica ainda 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.

Por ser uma licença restrita, como visto anteriormente, tais restrições têm o objetivo de garantir que o programa e seus tarefas derivadas sejam sempre redistribuídos sob os mesmos termos, garantindo a liberdade dos usuários. Em outras palavras, pode-se falar que a GPL permite aos usuários modificarem qualquer coisa no programa, exceto impor restrições adicionais a outros usuários, quando redistribuídos. É o conceito da melhoria contínua do software, já citada anteriormente.

Como essa licença se aplica a qualquer projeto derivado do original, ou seja, com partes do original, acaba que essa definição de derivação 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 em raras situações pode dar origem a controvérsias quanto à necessidade de reciprocidade em alguns usos atípicos do software.

Interessante notar e válido ressaltar que está no texto que, “caso não seja possível cumprir alguma decisão judicial ou lei local e ao mesmo tempo seguir os termos da licença, então não é permitido que o programa seja distribuído nessas circunstâncias ou nesse contexto.”

Em outras palavras, não basta simplesmente cumprir as regras tão somente da licença, precisa ter conformidade com as leis de direito autorais previstas em cada país onde o software precisa ser licenciado.

Por fim, os autores SABINO; KON(2009) citam que:

(..) *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 utilizar. 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 aberta essa possibilidade. **Um exemplo importante dessa escolha é o núcleo do Linux.**(grifo meu)

Abaixo veja, puramente por questões de espaço, uma parte da licença GPL ³, disponibilizada, na íntegra, em inglês, no site oficial da OSI. É possível também acessar uma versão traduzida não-oficial da GPL no site da GNU⁴:

GNU General Public License version 3

License Copyright: Copyright © 2007 Free Software Foundation, Inc.

License License: "Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed." (www.gnu.org/licenses/gpl-3.0.en.html).

License Contact: Free Software Foundation (fsf.org).

SPDX short identifier: `GPL-3.0-only`

[Further resources...](#)



Begin license text.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities

Imagem 5: Texto original, em parte, dos termos da Licença GPL.

³ Disponível em: <https://opensource.org/licenses/GPL-3.0>

⁴ Acesse o site com as traduções da GPL em: <https://www.gnu.org/licenses/translations.html>

Algumas considerações sobre as versões das Licenças GPL

A GPL 2.0 foi redigida em junho de 1991 e foram feitas pequenas modificações na licença. Essa versão manteve-se até 2007, quando saiu a GPLv3. A GPL 2.0 inclui um preâmbulo com os princípios e objetivos desta licença.

Já a GPLv3, estabelecida 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*. Se você encontrou um código e queria incorporá-lo em um projeto sob GPL, a GPL 2.0 dizia que a licença no outro código não tinha permissão para ter restrições que ainda não estavam na GPL 2.0. Sendo este o caso, dizíamos que a licença era compatível com GPL. No entanto, algumas licenças possuíam requisitos que não eram realmente restritivos, porque eram tão fáceis de cumprir. Por exemplo, algumas licenças dizem que não lhe dão permissão para usar certas marcas registradas. Essa não é realmente uma restrição adicional: se essa cláusula não existisse, você ainda não teria permissão para usar a marca registrada.

Agora, a GPLv3 fornece explicitamente a todos a permissão para usar um código que tenha requisitos como este. Esses novos termos devem ajudar a esclarecer os mal-entendidos sobre quais licenças são compatíveis com licenças GPL.

A GPL 2.0 cita muito sobre “distribution” (distribuição) – quando se compartilha o programa com outra pessoa, acaba que ocorre tal distribuição. No entanto, mais tarde observa-se que as leis de direitos autorais em outros países usam a mesma palavra distribuição, mas dão significados diferentes. Portanto, essas ambiguidades, inclusive na tradução, necessitava de reparos e esses acertos, depois de muita discussão gerou a versão atual.

Com isso, na GPLv3 que usa um novo termo, “convey” (transmitir) e fornece uma definição para esse termo no seu texto. “Convey” tem o mesmo significado que “distribute” (distribuir), mas agora que isso é explicado diretamente na licença, portanto deixando mais fácil para as pessoas, em todos os lugares, entenderem. Existem outras alterações menores em toda a licença, que também ajudarão a garantir sua aplicação em todo o mundo.

SABINO. KON(2009), indica que:

Outra mudança na GPLv3 é **relacionada a mecanismos de DRM, ou Digital Rights Management**. É sabido que a *Free Software Foundation* é contra o uso de DRM. Ainda assim, ela não quis impedir que software livre fosse utilizado para implementá-lo, já que isso limitaria a liberdade dos usuários. Como alternativa, decidiram limitar o impacto do tratado de *copyright* da World Intellectual Property Organization (WIPO), adotado em 20 de dezembro de 1996 sobre o assunto. (grifo meu)

E, como um todo, todas essas atualizações representam algo mais, a feitura de um *copyleft* melhor. São mudanças que vem para proteger mais ainda a liberdade dos usuários, incluindo permitir uma maior cooperação na comunidade de software livre. Mas atualizar a licença é apenas parte do trabalho, isto é, mudanças podem gerar alguns retrabalhos na adaptação da licença GPL 2.0 para a nova versão. Contudo, a GPLv3 oferece muitos benefícios e, sendo assim, os desenvolvedores também precisam usá-la em seus projetos. Ao lançar seu próprio software sob a nova licença, os usuários, outros desenvolvedores, distribuidores e até advogados se beneficiarão.

Vantagens e desvantagens

Vantagens:

- Como é bastante conhecida, sendo a licença mais utilizada em projetos de software livre, torna-se uma vantagem ter mais aporte de informações e versão mais atualizada de seu texto;
- A GPL, com seu intuito de ser incremental, cada adição de software por meio de contribuições da comunidade (terceiros) sendo que cada melhorias feitas ao software devem manter-se livres para poderem ser distribuídas.
- A GPL, por ser restritiva, também é usada frequentemente em um modelo comercial de licenciamento duplo. Ou seja, é possível fornecer algum software sob a licença GPL, conseguindo as benesses relacionadas ao software livre, mas ao mesmo tempo pode ser possível disponibilizar o software sob alguma outra licença que não imponha as restrições presentes na GPL. Assim, empresas que têm interesse em usar o software em licenças fechadas podem obter uma licença alternativa, normalmente pagando um determinado valor para a empresa detentora dos direitos sobre o software e redistribuir com o licenciamento duplo.

Desvantagens:

- 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.
- Vale frisar que 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.

A LICENÇA AGPL

A AGPL é uma adaptação da GPL que inclui um termo sobre uso de um software através de uma rede. Inicialmente ela foi escrita pela empresa Affero Inc. mas em 2007 foi lançada a AGPLv3 como parte do grupo de licenças da *Free Software Foundation*, que a absorveu.

Dessa forma, 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 licenciamento múltiplo. Como se vê, a AGPL é muito semelhante à GPL, mas mais onerosa. A GPL obriga a distribuir seu próprio código-fonte valioso se você distribuir arquivos binários. A AGPL dá um passo a mais, obrigando você a disponibilizar ao mundo o seu código-fonte se você o estiver usando em um servidor conectado à rede como um Software como Serviço (SaaS) ou como um Fornecedor de Serviços de Aplicativos (ASP).

Abaixo veja, também por questões de espaço, uma parte da licença AGPL⁵, disponibilizada, na íntegra, em inglês, no site oficial da OSI. É possível também acessar uma versão traduzida não-oficial da AGPL no site da GNU⁶:

GNU Affero General Public License version 3

[View Summary of GNU Affero General Public License v3 \(AGPL-3.0\) on TLDRLegal » \(Disclaimer\)](#)

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

Imagem 6: Texto original, em parte, dos termos da Licença AGPL.

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*.

⁵ Disponível em: <https://opensource.org/licenses/GPL-3.0>

⁶ Acesse o site com as traduções da AGPL em: <http://licencas.softwarelivre.org/agpl-3.0.pt-br.html>

CONCLUSÕES

Tomando como base toda a informação adquirida por este estudo, conclui-se que o software livre é a melhor opção, ou seja, torna-se inviável o uso de licenças fechadas para a distribuição do CEUB-OS.

Pode-se inferir, por meio do estudo, que qualquer licença, desde que de software livre pode ser indicado para se usar como licença do sistema operacional do UniCeub. Como a intenção maior seja a aplicação em ambientes de rede e com interligação de unidades da universidade, recomenda-se o uso da licença AGPL na versão mais atual, qual seja, AGPLv3.

Mais uma vantagem das licenças GPL, incluindo a AGPL é o uso mais intenso do *copyleft*,

Outro ponto convergente é a facilidade de escrita e aproveitamento das licenças de softwares livres, gerando a agilidade que o sistema desenvolvido proporciona, podendo ser customizado da melhor forma que o UniCeub necessita para seu sistema CEUB-OS.

REFERÊNCIAS BIBLIOGRÁFICAS

GUTIERREZ, Regina Maria Vinhais; ALEXANDRE, Patrícia Vieira Machado. Complexo eletrônico: introdução ao software. BNDES Setorial, Rio de Janeiro, n. 20, p. 3-76, set. 2004

PIMENTEL, Luiz Otávio; SILVA, Cláudio Eduardo Regis de Figueiredo e. Conceito jurídico de software, padrão proprietário e livre: políticas públicas. Sequência (Florianópolis), Florianópolis , n. 68, p. 291-329, June 2014 . Available from <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2177-70552014000100013&lng=en&nrm=iso>. access on 01 Oct. 2020. <https://doi.org/10.5007/2177-7055.2013v35n68p291>.

SABINO, Vanessa; KON, Fabio. Licenças de software livre história e características. 2009.

_____ <https://blog-br.softwareone.com/licenca-de-software>

_____ <https://www.gnu.org/licenses/quick-guide-gplv3.pt-br.html>