

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**ANDRÉ INÁCIO STRAGINSKI CARMONA**

**O SOFTWARE LIVRE NO LIMITE DA PROPRIEDADE  
INTELLECTUAL:**

**UMA BREVE APRESENTAÇÃO**

**Florianópolis, 2008.**

**ANDRÉ INÁCIO STRAGINSKI CARMONA**

**O SOFTWARE LIVRE NO LIMITE DA PROPRIEDADE  
INTELLECTUAL:**

**UMA BREVE APRESENTAÇÃO**

**Monografia submetida ao Departamento de  
Ciências Econômicas para obtenção da carga  
horária na disciplina CNM 5420 – Monografia.**

**Orientador: Prof. Dr. Armando Melo Lisboa**

**Florianópolis, 2008.**

## AGRADECIMENTOS

Ao esforço alheio de milhões e milhões de compatriotas anônimos que subsidiaram o meu direito de estudar numa universidade pública e de qualidade. A todos os meus professores, que de uma maneira ou outra contribuíram para o meu desenvolvimento intelectual. Em especial ao professor Armando, em cujo exemplo aprendi que a economia pode ser um campo menos árido e desprovido de vida que as abstrações e matematizações levam a crer.

Aos meus colegas e amigos de graduação, com cuja vivência aprendi as virtudes do bom debate.

À minha irmã, Marina, cuja inabalável paciência resistiu até mesmo às mais grosseiras brincadeiras.

Aos meus pais, Osny e Neivone, cujo amor incondicional e grandeza de espírito me guiaram em meio a um mundo nem sempre tão virtuoso.

*“Quem é cada um de nós senão uma combinatória de experiências, de informações, de leituras, de imaginações?”*

**(Italo Calvino)**

*“Alimento as vossas mentes quando falo. Repartiríeis entre vós o alimento, se o que trouxesse fosse para os vossos estômagos, e não chegaria inteiro a cada um; mas quantos mais fôsseis, em tanto maior número de pedaços dividiríeis o que vos desse. Aquilo que digo é inteiro para todos e inteiro para cada um.”*

**(Santo Agostinho)**

## RESUMO

Este trabalho visa apresentar o tema do software livre, articulando seus elementos com a temática da propriedade intelectual. Para tanto, caracteriza os aspectos principais do software livre, procurando definir o conceito de software livre, o propósito e modo das licenças sob as quais operam estes softwares, assim como o conceito de copyleft e o modelo de desenvolvimento que surge desse ambiente. Apresenta a discussão sobre propriedade intelectual, aponta o *trade-off* implícito nesse sistema de proteção, sugere a idéia de commons e enfatiza o caráter instrumental do copyleft quando interpretado nesses moldes. Por fim, coloca a proposta de cultura livre como a expansão das idéias do software livre para o âmbito da produção cultural.

**Palavras-chave:** 1. Software Livre  
2. Propriedade Intelectual  
3. Copyleft

## SUMÁRIO

<b>RESUMO .....</b>	<b>v</b>
<b>1. INTRODUÇÃO .....</b>	<b>7</b>
<b>1.1 Introdução .....</b>	<b>7</b>
<b>1.2 Objetivos .....</b>	<b>8</b>
1.2.1 Objetivo Geral .....	8
1.2.2 Objetivos Específicos .....	8
<b>1.3 Metodologia .....</b>	<b>8</b>
<b>2. SOFTWARE LIVRE .....</b>	<b>10</b>
<b>2.1 Definição implícita de software .....</b>	<b>10</b>
<b>2.2 Definição de software livre, software de código aberto e software proprietário .....</b>	<b>11</b>
<b>2.3 A história do software livre: do Unix ao GNU, e do GNU ao Linux .....</b>	<b>14</b>
<b>2.4 Copyleft e licenças de software livre .....</b>	<b>18</b>
2.4.1 Licença GPL (GNU) .....	20
2.4.2 Licença BSD .....	21
<b>2.5 Desenvolvimento de software livre .....</b>	<b>22</b>
<b>3. PROPRIEDADE INTELECTUAL E CULTURA LIVRE .....</b>	<b>27</b>
<b>3.1 Propósito e breve histórico da propriedade intelectual .....</b>	<b>27</b>
<b>3.2 Copyleft e commons .....</b>	<b>32</b>
<b>3.3 Creative commons e cultura livre .....</b>	<b>35</b>
<b>4. CONSIDERAÇÕES FINAIS .....</b>	<b>39</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>42</b>

# 1. INTRODUÇÃO

## 1.1 Introdução

O software livre tem sido, nos últimos anos, objeto de atenção por parte de um grande número de atores sociais. Acadêmicos, empresas, mídia especializada e governos têm observado atentamente os desdobramentos que este fenômeno exerce não somente sobre o seu campo original de atuação, a esfera do software, mas também na forma sutil pela qual o seu exemplo tem se irradiado para diversos debates.

O escopo do software livre usualmente é associado ao intenso desenvolvimento tecnológico que as últimas décadas trouxeram, especialmente na área denominada tecnologia da informação. Embora abundem teses e pesquisas sobre as novas relações que marcam a interação economia, sociedade e tecnologia neste início de século, são relativamente poucos os esforços que visam apresentar ou organizar a temática do software livre. É certo que o caráter multifacetado deste fenômeno dificulta essas propostas, no entanto, a relevância do tema tende a recompensar aqueles que enveredam por seus caminhos.

O software livre é, além de um conceito, um movimento social. As numerosas e variadas comunidades que se aglutinam ao redor dos projetos de desenvolvimento constituem prova do seu alcance. Não menos importante, a participação do software livre em diversos mercados de software, embora não seja possível apontar uma estatística geral, apresenta uma clara tendência de crescimento.

A atenção por parte de governos revela que o software livre também pode inspirar políticas públicas. A questão da independência tecnológica, por exemplo, é normalmente apontada como um dos campos onde o software livre pode desempenhar um importante papel.

Nessa linha, a crescente relevância que a discussão sobre direitos autorais enseja, constitui uma área ainda pouco explorada dentre as vastas possibilidades de imbricamento que essa temática oferece. De fato, o tema do copyleft participa de forma ainda muito tímida na construção de uma esquematização mais acessível e didática destas questões tão próprias do capitalismo contemporâneo.

No intuito de preencher infimamente esta lacuna, o presente trabalho se lança nessa empreitada.

## **1.2 Objetivos**

### **1.2.1 Objetivo Geral**

O objetivo geral proposto consiste em apresentar o fenômeno do software livre, articulando os seus elementos com a temática da propriedade intelectual.

### **1.2.2 Objetivos Específicos**

- λ Definir software livre e descrever sua construção histórica.
- λ Definir o modelo de desenvolvimento do software livre.
- λ Apresentar a temática da propriedade intelectual à vista das perplexidades que a cercam.

## **1.3 Metodologia**

O delineamento recente do objeto como fenômeno, em coaduno com o seu irreduzível caráter interdisciplinar ensejam uma irremediável opacidade na definição dos pressupostos teóricos. Nessas condições, a pesquisa exploratória é apontada como a metodologia mais adequada para se levar a bom termo os objetivos propostos. As investigações desta natureza buscam aproximar o pesquisador do fenômeno, no intuito de que este se familiarize com as características e peculiaridades do tema a ser explorado. Os elementos assim obtidos servem de subsídio para uma maior acuidade na percepção do tema.

Nesses termos, a proposta de procedimento metodológico inclui um esforço de pesquisa bibliográfica. A principal vantagem dessa ferramenta reside na possibilidade que o pesquisador tem de aglutinar uma gama de diferentes contribuições científicas disponíveis sobre o tema. A leitura e esquematização de diferentes fontes textuais constituem a prática básica dessa opção, assim, a base científica deste trabalho produz-se com o auxílio de bibliografias disponíveis em livros, artigos e sítios eletrônicos.

A consecução dos objetivos propostos demanda a organização deste trabalho, incluindo esta introdução, em três capítulos e uma conclusão.



No primeiro capítulo é apresentado o tema, o objetivo geral, os objetivos específicos e a metodologia.

O segundo capítulo congrega os principais elementos descritivos do software livre. Procura-se percorrer uma trajetória tal que a definição de software livre, juntamente com seu histórico, permita um melhor entendimento de duas das mais relevantes expressões do fenômeno: os modelos de licença e desenvolvimento de software livre.

O terceiro capítulo expõe a organização da temática da propriedade intelectual, articulando nas seções subseqüentes os possíveis significados do copyleft e a idéia de commons, assim como entendimento da noção de cultura livre.

Por fim, serão apresentadas as considerações finais do estudo e uma lista das referências bibliográficas utilizadas para composição deste trabalho.

## 2. SOFTWARE LIVRE

### 2.1 Definição implícita de software

Sinteticamente, define-se software, ou programa de computador, como uma seqüência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de uma informação ou acontecimento. No entanto, na condição de um artefato humano, o software é uma entidade que não se enquadra em definições convencionais. Fernandes (2003) o descreve como um elemento “de natureza mecânica, descritiva, complexamente hierarquizada, cognitivo-linguística e histórica.”

Certamente, uma interpretação mais usual utiliza o critério da utilidade ou finalidade, assim, software é o que torna o hardware<sup>1</sup> útil. Nesse caso, software também é produto e simultaneamente veículo para entrega do produto: nessa primeira forma, ele realiza o potencial de computação presente nos mais diversos dispositivos e atua como uma máquina capaz de transformar informação. Como veículo, age controlando os mais diversos dispositivos computacionais, operacionalizando redes de computador e viabilizando a criação de outros produtos, inclusive a ele iguais ou similares (BORGES, JAMBEIRO, SANTOS, 2003).

A fim de encaminhar uma outra percepção, propomos o entendimento de que software não é, de fato, uma máquina<sup>2</sup>, mas sim uma descrição de máquina. Ou seja, software é um artefato virtual, incapaz de realizar trabalho a menos que exista uma máquina que carregue e interprete as instruções e informações contidas no mesmo. Software, portanto, é uma meta-máquina.

Máquinas são extensões do ser humano. São mídias através das quais se estabelecem comunicações com resultados úteis e previsíveis. A natureza das linguagens de comunicação usuário-máquina permeia profundamente toda a relação e história do homem e dos artefatos que constrói, possuindo um impacto profundo sobre as atividades produtivas da sociedade. Criar linguagens está, portanto, no cerne da ação humana, e a prática do software permite o exercício desse processo criativo de forma eficiente e reproduzível (nos milhões de sistemas de computador que existem) como jamais se viu na história da humanidade. A definição da linguagem verbalizada por uma máquina computável é um processo criativo e evolutivo, baseado em

- 
- 1 Num computador, essa é a parte física propriamente, ou seja, a máquina em si, que é dividida em vários componentes. O hardware se distingue dos dados que o computador opera e do software que fornece instruções, baseadas nos dados, para o hardware (Dipold, 2005).
  - 2 Uma máquina é todo dispositivo mecânico ou orgânico que executa um trabalho, precisando para isso de uma fonte de energia (BRETON, 1991).

experiências cognitivos-coletivas (Fernandes, 2003).

É esta capacidade própria de criar linguagem, de viabilizar significados, que delinea, em muitos aspectos, o software. É nesse âmbito que ele se aproxima daquilo que nós usualmente consideramos como um bem intelectual. Se tomarmos como exemplo o programa IRPF (Imposto de Renda Pessoa Física) – desenvolvido pela Secretaria da Receita Federal -, o conjunto de formulários, tabelas de contribuição, mecanismos de cálculos e afins que descrevem a finalidade do software, constituem o que se pode chamar de “linguagem IRPF”. Como um exercício de definição de linguagem, é com a gramática (sintaxe) e lógica (semântica) bem definidas que uma virtual utilidade, atribuída a um projeto de software, pode ser bem sucedida. Nestes termos, Fernandes (2003) assevera: “o maior diferencial qualitativo do computador, relativo a todas as outras máquinas criadas pelo homem, é a capacidade de manipulação de representações simbólicas e discretas, estruturadas na forma de linguagens computáveis”.

## **2.2 Conceitos de software livre, software de código aberto e software proprietário**

Entende-se por software livre (*free software*) todo software cujo esquema de licenciamento permite ao seu usuário as condições de uso, reprodução, alteração e distribuição do seu código fonte, seja na sua forma original ou modificada, seja com ou sem custos. É crucial não confundir a liberdade proveniente dessas quatro condições com a potencial gratuidade de um software; existem programas que podem ser obtidos gratuitamente, mas que não podem ser redistribuídos nem modificados. Assim, software livre são ativos que podem ou não ser monetizados e transacionados nos mercados, dependendo da situação. Sobre este elemento de confusão, Richard Stallman<sup>3</sup>, fundador da *Free Software Foundation* (FSF<sup>4</sup>), esclarece: “Free software is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer’” (STALLMAN, 2001).

A propósito do surgimento do movimento pelo software livre no início de 1980, a história conta que Stallman, ao ver seu trabalho interrompido por uma impressora quebrada, solicitou ao fabricante que liberasse o acesso ao código-fonte. Por ocasião de uma resposta

---

<sup>3</sup> Richard Matthew Stallman, físico formado pela Universidade de Harvard em 1974, é um dos mais famosos, senão o mais famoso, ativista pelo software livre.

<sup>4</sup> <<http://www.fsf.org/>>

negativa, Stallman passou a pensar em uma forma de tornar acessíveis os programas e códigos-fontes. Este esforço está hoje concretizado no que a FSF coloca como os quatro conceitos básicos da filosofia do software livre:

- λ Liberdade de executar o programa, para qualquer propósito.
- λ Liberdade de estudar como o programa funciona, e adaptá-lo para suas necessidades (Acesso ao código-fonte é pré-requisito).
- λ Liberdade de redistribuir cópias do programa.
- λ Liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie.

Um programa então, é considerado livre se os usuários dispõem de todas essas liberdades. Nestes termos, a possibilidade de acesso ao código-fonte é condição imprescindível para a efetivação dos conceitos de uso e modificação do programa. Além do código-fonte, o autor (ou autores) do software outorga a liberdade para que outros programadores possam modificar o código original e redistribuir versões modificadas (HEXSEL, 2003).

É importante ressaltar que há diferença entre o software livre e o software de código aberto. O termo código aberto (*open source*) é, em princípio, uma categoria que enfatiza apenas a abertura do código-fonte de um software. Seria portanto um conceito diferente de software livre porque as premissas de liberdade não necessariamente devem ser observadas (SOFTEX, 2005).

A *Open Source Initiative* (OSI<sup>5</sup>) lista o seguinte conjunto de critérios como qualificadores necessários de um modelo de licenciamento tipo código aberto:

- λ Distribuição livre, sem pagamento de royalties ou semelhantes.
- λ Código fonte deve sempre estar aberto.
- λ Permitir modificações e trabalhos derivados.
- λ Garantir integridade autoral do código fonte.
- λ Não discriminar pessoas ou grupos.
- λ Não discriminar áreas de conhecimento, setores ou atividades.
- λ Direitos de licença redistribuídos sem necessidade de licenças adicionais pelas

---

5 <<http://www.opensource.org/docs/osd>>

partes.

- λ A licença não deve ser ligada a um produto específico.
- λ A licença não pode restringir outros softwares que são divulgados conjuntamente.<sup>6</sup>

A despeito dos princípios da OSI serem muitos parecidos com os preconizados pela licença mais comum no âmbito do software livre - a GPL -, a ênfase nos direitos autorais e a não restrição, na ponta, de fechamento do código constituem talvez o principal diferencial do software de código aberto. Enquanto as idéias de software livre estão mais vinculadas às questões de perpetuação das quatro liberdades, as de código aberto se colocam mais próximas das rotinas e práticas de produção e negócio, como a agilização do desenvolvimento do software através de comunidades abertas (SOFTEX, 2005). O proeminente caráter ideológico impingido pela FSF, alega a OSI, obscurece o principal argumento por um software de código aberto. A idéia central é que, em um modelo aberto de desenvolvimento, quando os programadores podem ler, modificar e distribuir o código, naturalmente surgem melhorias, adaptações e correções que têm como consequência a evolução mais eficiente - tanto técnica quanto econômica - do programa. De certa forma, a OSI constitui um movimento à parte do software livre. A respeito desta discussão, Stallman (2001) afirma:

Nearly all open source software is free software; the two terms describe almost the same category of software. But they stand for views based on fundamentally different values. Open source is a development methodology; free software is a social movement. For the free software movement, free software is an ethical imperative, because only free software respects the users' freedom. By contrast, the philosophy of open source considers issues in terms of how to make software "better"—in a practical sense only.

Por fim, posicionado no lado oposto do espectro de licenças, o software proprietário é definido como aquele cuja licença de uso proíbe em alguma medida a cópia, redistribuição ou modificação do programa por parte do usuário. Normalmente a redistribuição é ilícita e o código-fonte fechado; ademais, é bastante usual a cobrança de um valor por cópia utilizada.

É relevante esclarecer que “comercial” e “proprietário” não são necessariamente a mesma coisa; software comercial é o software desenvolvido por um empresa com o objetivo de lucrar com sua utilização. A maioria dos softwares comerciais é proprietária mas existe software livre que é comercial, e existe software simultaneamente não-livre e não-comercial (HEXSEL, 2003).

---

<sup>6</sup> Conforme traduzido e resumido por SOFTEX (2005).

## 2.3 A história do software livre: do Unix ao GNU, e do GNU ao Linux<sup>7</sup>

O surgimento do software livre está associado ao desenvolvimento da informática, principalmente nas décadas de 1960 e 1970. À época, a diferenciação entre software e hardware inexistia ou pouco importava, dada a considerável maior importância do equipamento físico, em detrimento da programação; até então a indústria fornecia o código fonte conjuntamente com o hardware e o software livre era regra, não exceção.

Dez anos mais tarde, as aplicações e o número de usuários aumentaram a tal ponto que passou ser possível comercializar o software ao invés de distribuí-lo gratuitamente. A situação se inverteu, de forma que o software tornou-se relativamente mais importante que o hardware e os fabricantes passaram a vender os programas sem acesso ao código fonte e com crescentes restrições a redistribuições (HEXSEL, 2003).

A história do software livre que se propõe neste trabalho começa com o desenvolvimento do sistema operacional<sup>8</sup> Unix<sup>9</sup> em 1969. A sua criação se deu nos laboratórios Bell, então pertencentes à estadunidense *American Telephone and Telegraph* (AT&T), que à época ocupava posição monopolística no mercado de telecomunicações norte americano.

O Unix, por sua facilidade de uso, controle e elegância de projeto, impressionou positivamente os demais pesquisadores dos laboratórios Bell e várias universidades, que obtiveram cópias do sistema para uso próprio. Como à época a AT&T podia atuar somente no mercado de telecomunicações, o sistema foi concedido às universidades, estimulando a distribuição do seu código fonte de modo a fomentar inovações. A partir de então, encontros científicos em torno do Unix se criaram e novas idéias e melhoramentos foram rapidamente incorporados ao sistema. Em pouco tempo, o Unix tornou-se um ponto de referência para universidades e centros de pesquisa norte-americanos.

Uma das instituições que mais ativamente contribuiu para este processo foi a Universidade de *Berkeley*, que começou a utilizar o Unix desde a sua criação em 1969. Cerca de oito anos depois, em 1977, foi criada a *Berkeley Software Distribution* (BSD), organização vinculada à universidade, que tinha por função distribuir a versão Berkeley do sistema

---

7 Este histórico do software livre, do UNIX, do GNU e do Linux aqui apresentado baseia-se nos trabalhos de Dipold (2005) - com base em Saleh (2004) - e Bacic (2003) - com base em Ascenso e Santos (2003), Hauben (2003), Lotterman (2001), Stallman (1996), Tenenbaum (1994) e UnixSupport (2003).

8 Sistema operacional é o programa básico que controla o hardware do computador e permite execuções dos demais aplicativos. Alguns exemplos: Windows, GNU/Linux, MacOS, DOS.

9 O nome Unix provem de UNICS (*Uniplexed Information and Computing Services*), sistema operacional derivado do MULTICS (*Multiplexed Information and Computing Services*), projeto desenvolvido pelos laboratórios Bell em 1964 (Hauben, 2003).

operacional juntamente com editores de texto e compiladores<sup>10</sup>.

A última versão do Unix conduzida pelos laboratórios *Bell* foi liberada em 1975. Após este ano, o sistema operacional passou a ser controlado pelo *Unix System Laboratories* (USL), a recém criada filial dos laboratórios *Bell*, que agora passava a atuar com objetivos comerciais. À Universidade de Berkeley, devido à sua vasta experiência no desenvolvimento do Unix, coube assumir o papel antes desempenhado pela sucursal da AT&T. Com uma participação dessa magnitude e um projeto de caráter aberto, o BSD obteve grande projeção e aumentou consideravelmente o seu número de adeptos. Todas as distribuições de programas binários eram sempre acompanhadas de código fonte, assim o usuário do sistema tornava-se não somente um utilizador passivo, mas sim um colaborador, uma vez que tinha a possibilidade de criar novas funcionalidades e melhorar as que já existiam. A versão Unix original, que agora pertencia à USL e tinha o seu código fechado, dependia exclusivamente de sua equipe interna e de seus recursos próprios para o desenvolvimento. O resultado foi que no ano de 1989, o BSD 4.3 gozava de uma popularidade e de uma reputação muito maiores que o *Unix System V*<sup>11</sup> da USL.

No entanto, permaneciam certas restrições para se utilizar a versão BSD, pois ainda era necessário obter junto à AT&T/USL a permissão de uso para partes de código por ela desenvolvidas. Iniciou-se então um movimento para separar o código desenvolvido pela AT&T, que era proprietário, do código desenvolvido por *Berkeley*, que era aberto. Dessa iniciativa nasceu, em 1989, a distribuição chamada *Networking Release 1*, que embora não fosse um sistema operacional completo, possuía quase a totalidade do seu código disponível.

O lançamento do *Networking Release 1* obteve grande aceitação e incentivou os desenvolvedores do BSD a promoverem a eliminação de todo o código proprietário. Em 1991 somente seis arquivos no sistema se qualificavam como de licença proprietária, no entanto, o seu desenvolvimento demandava um alto grau de complexidade; resolveu-se, assim, organizar um nova versão chamada *Networking Release 2*. Esse lançamento provocou um grande furor na comunidade usuária e em apenas seis meses os substitutos para os arquivos que faltavam estavam prontos e integrados à distribuição principal.

No começo de 1990, para desenvolver e distribuir comercialmente uma versão do código BSD/Unix, a BSD criou a *BSD Inc.*, empresa que ficaria incumbida de administrar esta opção comercial. A USL reagiu imediatamente contra a universidade de *Berkeley*,

---

10 Um compilador é um programa que transforma um código escrito em uma linguagem inteligível pelo homem (código fonte) em uma programação compreensível pelo hardware (código objeto).

11 O *Unix System V* foi a primeira versão a padronizar muitos dos elementos do UNIX. O primeiro lançamento foi feito em 1989.

alegando que os produtos da *BSD Inc.* ainda continham códigos de sua propriedade. A contenda resultou em um processo judicial e, durante dois anos, de 1992 a 1994, qualquer sistema operacional que tivesse como base o *Networking Release 2* foi proibido de ser distribuído. Ao final, *Berkeley* foi condenada a remover três arquivos de um total de mil e oitocentos integrantes do sistema, onde, no entender da justiça, existiam códigos que eram propriedade da USL. Em pouco tempo esses arquivos foram reescritos.<sup>12</sup>

Ainda no ano de 1984, Richard Stallman - que à época trabalhava como pesquisador no *Massachusetts Institute of Technology* (MIT) - decidiu criar um sistema operacional fundamentando nos preceitos que mais tarde viriam a compor os quatro conceitos básicos do software livre. Stallman almejava organizar uma comunidade de desenvolvedores que trabalhassem norteados pelos princípios da cooperação e do compartilhamento, espírito este que prevalecia nos primórdios da informática e que tornava imprescindível que o sistema possuísse código fonte aberto. O desenvolvimento de um software deveria ocorrer de forma evolucionária, na qual um programador pudesse, a partir de um programa já existente, adicionar melhorias, novas funcionalidades, novos recursos e até mesmo criar novos aplicativos sem a necessidade de se começar do zero.

Optou-se por produzir um sistema operacional fundamentado pelo Unix, pois dessa forma a migração dos usuários para o novo sistema seria facilitada. O nome escolhido para o projeto foi GNU (*GNU is not Unix* - GNU não é Unix). Simultaneamente foi criado um novo tipo de licença, a *General Public License* (GPL), para salvaguardar e preservar o acesso ao código fonte.

A fim de economizar recursos, os desenvolvedores do GNU optaram por integrar programas que fossem livres, mesmo não que não fossem regidos pela licença GNU (GPL); exemplos disso são a utilização do sistema de janelas desenvolvido pelo MIT, o *X Windows* e o protocolo de comunicação TCP/IP, desenvolvido pelo departamento de defesa norte americano.

No mesmo ano, Stallman utilizou-se do laboratório de Inteligência Artificial do MIT para concretizar o desenvolvimento primordial do sistema GNU. Inicialmente foram criados o editor de texto Emacs e em seguida o compilador GCC (*GNU C Compiler*). Esses aplicativos eram distribuídos de duas maneiras: através do sítio do MIT na internet (na época de uso muito restrito) ou por intermédio da venda de fitas que continham os software

---

12 O BSD hoje se divide em basicamente três projetos: o NetBSD, que tem foco na portabilidade para diferentes plataformas; o FreeBSD, que procura desenvolver um sistema operacional geral, capaz de ser utilizado por pessoas com menos conhecimento técnicos; e o OpenBSD, que tem foco em segurança. Todos distribuídos como software livre.



gravados. A venda das fitas iniciou um dos primeiros negócios envolvendo software livre. O sucesso do Emacs deu visibilidade ao projeto GNU e estimulou o recrutamento de novos desenvolvedores.

Em 1985 é formalizado o movimento pelo software livre com a criação da FSF, à qual caberia a responsabilidade pelo desenvolvimento e manutenção do GNU, assim como a criação e implementação de novas modalidades de licenças para software livre.

Inicialmente a fundação desenvolveu, dentre outros aplicativos, o interpretador de comandos BASH (*Bourne Against Shell*), cujo nome provinha de um trocadilho com o seu equivalente no sistema Unix original, o *Bourne Shell*. Finalmente, após seis anos, em 1990, o sistema GNU estava praticamente completo, faltando somente a criação do núcleo (*kernel*). O núcleo original do sistema GNU, chamado Hurd, revelou-se de uma complexidade inesperada. Mesmo no presente momento, em 2008, o projeto utilizado - conhecido como *microkernel* - ainda não possui a estabilidade necessária para ser utilizado em ambientes que não sejam de teste e desenvolvimento.

Em 1991, Linus Torvalds, do Departamento de Ciências da Computação da Universidade de Helsinque, Finlândia, criou um núcleo compatível com o Unix (utilizando o GCC), baseado no Minix, um sistema operacional desenvolvido pelo professor Andy Tannenbaum para fins educacionais. Esse núcleo, no entanto, utilizava a arquitetura chamada de núcleo monolítico, que à época era considerada ultrapassada. Apesar disso, sua abordagem permitia um desenvolvimento mais fácil e eficiente. Após criar a versão 0.02 do núcleo, que recebeu o nome de Linux, Torvalds disponibilizou o código fonte na Internet, conclamando programadores de todo o mundo para que o ajudassem a aprimorar o programa.

Em 1992, o *kernel* Linux foi incorporado com sucesso ao sistema GNU, gerando um sistema operacional livre e completo denominado GNU/Linux, embora hoje seja mais conhecido apenas como Linux.

Com o surgimento do GNU/Linux como um sistema completo e imediatamente disponível, as atenções de desenvolvedores do mundo inteiro voltaram-se para ele, e houve um vigoroso incremento no número de interessados em seu uso e desenvolvimento. Em alguns poucos anos, o projeto GNU/Linux atingiu funcionalidades e estabilidade comparáveis a sistemas operacionais proprietários já consolidados.

Em 1996, tratando da relação entre o projeto GNU, o Linux e a filosofia do software livre, Richard Stallman escreve:

O projeto GNU não é somente desenvolvimento e distribuição de alguns softwares

livres úteis. O coração do projeto GNU é uma idéia: que software deve ser livre, e que a liberdade do usuário vale a pena ser defendida. Se as pessoas têm liberdade, mas não a apreciam conscientemente, não irão mantê-la por muito tempo. Se quisermos que a liberdade dure, precisamos chamar a atenção das pessoas para a liberdades que elas têm em programas livres.<sup>13</sup>

E acrescenta:

O método do projeto GNU é que programas livres e a idéia da liberdade dos usuários ajudem-se mutuamente. Nós desenvolvemos software GNU, e conforme as pessoas encontram programas GNU ou o sistema GNU e começam a usá-los, elas também pensarão sobre a filosofia GNU. O software mostra que a idéia funciona na prática. Algumas destas pessoas acabam concordando com a idéia, e então escrevem mais programas livres. Então, o software carrega a idéia, dissemina a idéia e cresce a idéia.<sup>14</sup>

O estrondoso sucesso do Linux foi apenas o começo de uma intrincada trajetória de avanços e percalços por parte do software livre. Outros aplicativos, nas mais diversas funções, atingiram grau de sucesso similar em seus respectivos nichos. Como exemplo, a maioria dos servidores de Internet do mundo roda em Apache<sup>15</sup> e uma significativa e crescente parcela de usuários usa o navegador Mozilla Firefox<sup>16</sup>, assim como o editor de textos Openoffice.<sup>17</sup>

## 2.4 Copyleft e licenças de software livre

O uso de software livre está, da mesma forma que o software proprietário, baseado em licenças. As regras de utilização, portanto, se materializam conforme o regime de licença estabelece uma ou outra propriedade no uso de um ou outro programa. O usuário deve concordar e aceitar a licença associada ao software para utilizar os códigos que ali estão. São as condições precisadas nestas licenças que definem se um software é considerado livre, de código aberto ou proprietário. Como já colocado, as licenças de software livre autorizam qualquer usuário a utilizar, copiar, modificar e distribuir o software, conforme determinadas

---

13 Stallman (2008).

14 Ibid.

15 O servidor Apache é o mais bem sucedido servidor de Internet do mundo. Foi criado em 1995 por Rob McCool.

16 O Mozilla Firefox é um navegador livre e multi-plataforma desenvolvido pela *Mozilla Foundation*. Atualmente destaca-se como a principal alternativa ao Microsoft Internet Explorer, navegador de código proprietário dominante no mercado. À ascensão do Firefox se deve o recrudescimento recente da chamada “guerra dos navegadores”.

17 O Openoffice é um conjunto livre e multi-plataforma de aplicativos para escritórios. O Openoffice é baseado em uma antiga versão do StarOffice, o StarOffice 5.1, adquirido pela Sun Microsystems em agosto de 1999. O código fonte foi liberado dando início a um projeto de desenvolvimento de um software de código aberto em 13 de outubro de 2000. O principal objetivo era fornecer uma alternativa de baixo custo e de alta qualidade.

regras. Em geral, as licenças de software proprietário permitem que o usuário somente utilize o programa de acordo com as regras do titular do software, sendo proibida sua reprodução, instalação múltipla, alteração, cessão, revenda ou distribuição sem o devido pagamento adicional.

A intenção da FSF, quando iniciou seu projeto, era proteger legalmente a comunidade do software livre, de forma que os códigos desenvolvidos não fossem apropriados por desenvolvedores de software proprietário. Para esse propósito, o simples uso do domínio público não era aceitável, pois implicaria na possibilidade de posse e conclamação de autoria por parte de algum ente privado.

A solução para esse problema veio na forma de uma criativa estratégia, concebida sobre a idéia de utilização da estrutura legal de proteção aos direitos autorais para garantir a liberdade dos usuários de computadores. Através das leis de copyright (direitos autorais), o autor tomaria posse da obra, garantindo assim sua autoria e seus direitos patrimoniais. Em seguida esse autor imporia condições para seu licenciamento, de forma que, ao invés de restringir o acesso do público e controlar as cópias e distribuições, daria ao licenciado a liberdade de usar, distribuir e modificar o programa original da forma que bem entendesse. Por fim, seria imposta uma restrição adicional: toda e qualquer obra derivada obrigatoriamente teria que fornecer ao licenciado os mesmos direitos. Esta última cláusula, fonte do chamado “efeito contaminação”, garantiria que, uma vez livre, o software deveria para sempre assim deveria permanecer. Ao preservar os direitos autorais, simultaneamente ao liberar os direitos comerciais, esta modalidade de licenciamento contraria frontalmente a forma usual de publicação restritiva no mercado. Esse recurso foi batizado como “copyleft”, em uma espirituosa paródia que inverte para esquerda o sentido de direita do copyright.

Cabe ressaltar que nem todas as licenças de software livre impõem o copyleft. Existem vários estilos de licenças que se distinguem pelo grau de liberdade outorgado ao usuário. De forma geral, todas as licenças compartilham das seguintes características:

- λ Proteger a identidade do autor;
- λ Distribuir os códigos fontes;
- λ Qualquer trabalho que inclua parte do software deve citar o autor;
- λ Instalação do software em um número irrestrito de computadores;
- λ Não restringir a comercialização do software;<sup>18</sup>

---

18 Bacic (2003).

Em alguns casos, a escolha da licença a ser utilizada é decisão do autor. Ele tanto pode optar por uma das licenças conhecidas ou escrever os termos de uma própria. Entretanto, se houver a utilização de linhas de código sobre efeito de outras licenças, possivelmente será necessário se adequar a essas outras regras. Apesar disso, o desenvolvedor, como autor, tem o direito de dar qualquer destino ao código por ele desenvolvido.

Existe, também, a possibilidade de licenciamento dual, ou em outras palavras, um mesmo código fonte pode ser liberado pelo autor sob duas (ou mais) licenças distintas, conforme o seu interesse. Numa situação hipotética, um desenvolvedor poderia disponibilizar o seu código fonte sob uma licença para ser utilizado em outros projetos de software livre, ao mesmo tempo em que disponibilizaria este mesmo código sob uma licença mais restritiva para fins de incorporação em um software proprietário. Este é o caso do banco de dados MySQL (SOFTEX, 2005).

A seguir, apresentamos duas das mais comuns licenças de software livre. A primeira se enquadra nos termos do copyleft ao passo que a segunda não.

#### **2.4.1 Licença GPL(GNU)**

Em 1989, com o conceito de copyleft já definido, foi criada, pela FSF, a primeira licença de software livre, chamada GNU *General Public License* (GPL). A GPL é a licença que acompanha os pacotes distribuídos pelo projeto GNU, e mais uma grande variedade de software, incluindo o núcleo do sistema operacional Linux. A formulação desta licença é a que mais se aproxima da perfeita efetivação das quatro liberdades postuladas pela FSF. Apesar disso, a GPL claramente permite e incentiva qualquer forma de uso empresarial dos recursos em software livre (tanto pela prestação de serviços, pela venda de mídia contendo software, pelo oferecimento de garantia e suporte técnico ou pelo desenvolvimento de novas funcionalidades) o que demonstra que o foco da licença é a liberdade, e não a gratuidade (SALEH, 2004).

Existe uma alternativa para que bibliotecas<sup>19</sup> do projeto GNU possam ser utilizadas em um software comercial, é a licença derivada GNU *Library General Public License* (LGPL). Por via dessa licença, as bibliotecas desenvolvidas pela GNU podem ser livremente utilizadas

---

<sup>19</sup> Rotinas executáveis dos softwares que podem ser executadas a parte do programa principal. Elas são utilizadas para que não seja necessário reescrever diferentes códigos para executar tarefas comuns. Quando quer utilizar estas funções, o programador insere as bibliotecas dentro do seu programa, fazendo com que o produto final seja a combinação do seu próprio código com o código da biblioteca utilizada.

em aplicações comerciais, tendo como única restrição que o programa seja capaz de aceitar as atualizações das bibliotecas livres. No entanto, essa licença não garante que o programa que esteja utilizando uma biblioteca provinda de um software livre também o seja. Por isso a FSF desencoraja fortemente o uso dessa licença, recomendando-a somente em casos muito específicos, onde não haja outra alternativa. Um exemplo é o Openoffice, o mais avançado conjunto de aplicativos de escritório existente para os sistemas operacionais GNU/Linux.

A GNU ainda possui uma licença especial para a documentação, a GNU *Free Documentation License* (FDL), que visa assegurar que documentos, manuais e livros sejam livres, sendo permitida a alteração, cópia e redistribuição de forma gratuita ou comercial. A FDL foi escrita originalmente para os manuais que acompanhassem o sistema operacional GNU. Entretanto, atualmente vários outros tipos de publicações, principalmente de cunho técnico sobre software livre, utilizam essa licença.

#### **2.4.2 Licença BSD**

A licença Berkeley Software Distribution (BSD) foi criada pela Universidade de Berkeley por ocasião do desenvolvimento de sua própria versão do sistema operacional Unix. Essa é uma licença considerada “permissiva” porque impõe poucas restrições sobre a forma de uso, alteração e redistribuição do software licenciado.

A sua estrutura, adjetivada como simples e enxuta, preconiza somente algumas poucas restrições: na redistribuição do código fonte deve ser mantido o aviso do copyright original para identificar o autor; as distribuições binárias, sem o código fonte, devem conter o copyright na documentação; o nome do autor não pode ser utilizado para endossar versões modificadas do programa; há a ressalva de que o programa pode não funcionar para qualquer finalidade e que seus autores não têm responsabilidade sobre o seu uso (SALEH, 2004).

De certa maneira, a licença BSD se enquadra mais adequadamente na proposta do movimento pelo código aberto do que nos preceitos do software livre. Como não há restrições, o código pode ser utilizado para qualquer finalidade, inclusive incorporação em software proprietário. O argumento implícito assevera que essa licença é “verdadeiramente” livre, pois não impõe qualquer restrição ao licenciado, mesmo que essa possibilidade leve a liberdade, digamos, “negativa” de fechamento do código. A FSF a considera prejudicial por não se preocupar com a questão central da proposta pelo software livre. Essa não é uma licença que adere ao conceito de copyleft, pois não garante o efeito contaminação.

Uma boa parte das licenças que existem atualmente têm como base a licença BSD e os conceitos do movimento pelo código aberto. Empresas adeptas do modelo de desenvolvimento híbrido (licenciamento dual), frequentemente utilizam essas licenças para fim de viabilizar o escopo entre uma versão proprietária e outra livre.

## 2.5 Desenvolvimento de software livre

É certamente no modelo de desenvolvimento do software livre que se encontra um dos mais robustos argumentos pela adoção da prática de livre acesso ao código fonte. O desenvolvimento usual de software proprietário ocorre por grupos de desenvolvedores organizados internamente a uma empresa, ou de empresas contratadas para tal, sob obrigações que impedem o uso e divulgação de informações diretamente relacionadas ao programa em desenvolvimento. O código escrito é considerado um ativo de vital importância, e como tal, todo o processo de desenvolvimento é envolto em uma bruma de sigilo industrial e proteção à propriedade intelectual. Nesse âmbito, a despeito da natureza peculiar do software, a prática tradicional de criação e produção se assemelha positivamente à de um bem industrial.

O software livre, por outro lado, permitiu o surgimento de inovadores modelos de desenvolvimento de software. Viabilizados e potencializados pela Internet, esses modelos são fundamentados em um novo vetor: a colaboração em rede de desenvolvedores. Assegurado o acesso ao código fonte, a morfologia da rede facilita exponencialmente a distribuição de conhecimentos, responsabilidades e melhores práticas para todos os participantes de uma determinada comunidade, sejam eles programadores, designers, tradutores ou simples usuários que cooperam com sugestões ou identificação de *bugs*<sup>20</sup>. A própria Internet, enquanto uma infra-estrutura de software e protocolos, é resultado de um modelo similar de desenvolvimento:

A cultura de engenharia praticada pela comunidade que desenvolve os protocolos empregados na Internet se baseia na publicação de propostas de protocolos ou serviços, e na avaliação destas propostas em listas de discussões via correio eletrônico. Geralmente, as propostas são acompanhadas de uma *implementação de referência*, cujo código fonte é disponibilizado a todos os interessados. De posse da especificação dos protocolos e do código que os implementa, a comunidade se lança à avaliação, correção de eventuais erros, discussão e aperfeiçoamento da proposta ou implementação. Após o período de discussão aberta ao público, os protocolos são avaliados e sacramentados na *Internet Engineering Task Force* (IETF), que é a

---

<sup>20</sup> Disfunção na estabilidade e funcionalidade, causada por um erro não previsto no código fonte de um software.

entidade responsável pela formalização e publicação dos protocolos.[...] A disponibilidade das propostas de protocolos, através dos Request For Comments (RFCs), permite a discussão e a depuração dos protocolos por uma grande comunidade de especialistas. Este processo tem produzido protocolos extremamente confiáveis, robustos e escaláveis, especialmente quando se considera que a Internet é heterogênea, multi-plataforma e geograficamente distribuída, e que opera a mais de trinta anos mantendo a compatibilidade entre várias gerações de tecnologia (Hexsel, 2003).

O desenvolvimento do *kernel* do Linux é outro exemplo amplamente comemorado como um modelo baseado em uma rede colaborativa de desenvolvedores. A dinâmica de operação destas comunidades desenvolvedoras depende fundamentalmente de um princípio mutualista, onde os participantes recebem dela mais do que nela depositam. O fluxo de contribuições, se bem organizado, enseja ganhos de sinergia que, via de regra, muito dificilmente seriam recriados em modelos mais tradicionais.

A mutualidade é o mesmo princípio que rege a comunidade científica e incentiva os pesquisadores a publicarem e divulgarem seus trabalhos. Sobre este notável paralelo, Hexsel (2003) comenta:

A publicação tem duas finalidades importantes: (1) os resultados ficam sujeitos ao escrutínio pela comunidade científica, o que garante sua qualidade e confiabilidade, e (2) estes resultados se incorporam ao corpo de conhecimento da área e ficam disponíveis para uso pelos próprios pesquisadores, estudantes e demais interessados. Este é o mesmo processo a que estão sujeitos os protocolos da Internet. Uma vez revisados e aprovados pela comunidade, os protocolos ficam disponíveis ao público para que sejam implementados e estudados, pela indústria, na academia, e pelas novas gerações de projetistas de protocolos. Da mesma forma, a distribuição do código fonte permite sua utilização por outros programadores em seus próprios projetos, bem como possibilita o aprendizado por outros programadores através do estudo do código dos programas que empregam diariamente. Sob esta ótica, o código fonte dos programas é uma forma de conhecimento científico.

Com a generalização dessas práticas, as implicações para o desenvolvimento de software livre foram marcantes. O lançamento de diversos projetos concretizou um número igual ou maior de comunidades. Em termos de estruturação, estas “redes” variam desde complexos sistemas auto-organizados a hierarquias com regras e níveis de acesso diversificado. O processo coletivo é orientado para diferentes formas de colaboração. Geralmente existe um núcleo de desenvolvedores composto por programadores ativos, imediatamente circundados por programadores esporádicos; outros apoiadores, como tradutores, artistas gráficos, editoras e empresas; e um grupo de usuários ativos e não-ativos. Embora o desenvolvimento de um software específico seja uma motivação poderosa, a relativa fragilidade e dispersão dos laços que os une se traduz em comunidades de desenvolvimento heterogêneas. A necessidade de algum grau de liderança promove a ação de

diferentes tipos de governanças.

Nessas condições, a dinâmica dos trabalhos requisita a atuação de líder(es) de projeto. Quando não há um indivíduo ou grupo que imponha algum grau de organização aos múltiplos desenvolvedores - seja pela prerrogativa da autoria original ou brilhantismo reconhecido -, um segundo princípio passa a operar: os indivíduos com atuação mais destacada, em conformidade com a aprovação da comunidade que se posiciona ao redor, passam a atuar decidindo quais colaborações serão implementadas na próxima versão e os rumos e prioridades do projeto. Existe nesse caso um critério de mérito que estabelece as responsabilidades. Nesses condições, a quantidade e qualidade do código contribuído, a participação ativa, opiniões e sugestões pertinentes usualmente qualificam e promovem as lideranças.

É importante ressaltar que, a despeito de uma organização razoavelmente bem definida e estruturada, o modelo de desenvolvimento de software livre aqui descrito - um coletivo de indivíduos regidos por uma vontade dispersa e em torno de um fluxo de contribuições -, quando posto em comparação à prática de desenvolvimento de software proprietário, ou mesmo em paralelo à criação e produção de bens industriais, pode aparentar uma total desordem. É a respeito dessa aparência anárquica que Eric Raymond, um dos mais influentes ativistas do movimento pelo software livre, em seu famoso artigo, *The Cathedral and the Bazaar*, trata quando denomina o modelo de desenvolvimento do software livre como “bazar”, em contraposição ao que ele define como o modelo de organização proprietário: “a catedral medieval”.

O estilo Linus Torvalds de desenvolvimento - libere cedo e frequentemente, delegue tudo que possa, esteja aberto ao ponto da promiscuidade - veio como uma surpresa. Nenhuma catedral calma e respeitosa aqui, ao invés, a comunidade Linux parece assemelhar-se a um grande e barulhento bazar de diferentes agendas e aproximações (adequadamente simbolizada pelos repositórios do Linux, que aceitaria submissões de qualquer pessoa) de onde um sistema coerente e estável poderia aparentemente emergir somente por uma sucessão de milagres (Raymond, 1997).

A singela analogia sugere um significativo e genial contraponto. Raymond, ao argumentar que o modelo tradicional empregado na indústria é similar ao projeto de uma “catedral medieval”, no qual um restrito número de projetistas exerce férreo controle sobre o trabalho de um pequeno exército de operários, descreve tacitamente qual seria a relação entre o comando do projeto (apontado, dentre outros “vícios”, como o departamento de marketing) e o grupo de programadores contratados que são submetidos a determinados prazos, metodologias e tarefas. Uma consequência bastante prejudicial desse direcionamento, dessa



rigidez, é a dificuldade que esse tipo de modelo possui para atingir massa crítica de usuários. Como “na catedral” os projetistas definem as características técnicas e funcionais do produto e repassam o desenvolvimento aos programadores, quando se atinge a fase de testes, somente alguns poucos testadores validam o produto, que é então liberado para uso e/ou comercialização. A limitação do teste e do *feedback* à um universo restrito de *beta-testers* (testadores) muitas vezes compromete a estabilidade e maturidade da versão inicial de lançamento, simplesmente porque o prazo estipulado expirou.

Sobre o modo de organização do software livre, semelhante a um bazar, onde aparentemente não há hierarquia entre os participantes e todos cooperam para que o bazar seja atrativo aos compradores, Raymond identifica uma organização informal, em que os projetos avançam somente se houver interesse por parte da comunidade. A relação entre a potencial criação de um novo programa e o grupo desenvolvedor é muito mais orgânica. Uma vez disponibilizado, o código é estudado e esmiuçado por qualquer um que tenha interesse, o que intensifica o processo de depuração, descoberta e correção de erros. Esse definidor facilita enormemente a obtenção de massa crítica nos projetos de software livre. A esse respeito, Raymond estabelece o que chama de “Lei de Linus”, uma de suas mais conhecidas afirmações: “Dada uma base grande o suficiente de *beta-testers* e co-desenvolvedores, praticamente todo o problema será caracterizado rapidamente e a solução será óbvia para alguém”<sup>21</sup>. Outra característica interessante é a possibilidade de se realizar alterações específicas, de acordo com as necessidades individuais de cada usuário, gerando versões personalizadas e que atendem perfeitamente cada característica demandada. Esta é uma das práticas com a qual é possível criar atividades comerciais com software livre: a personificação de versões.

Naturalmente, se um projeto não atrai a atenção de um número suficientemente grande de colaboradores é bastante provável que seu desenvolvimento pare. No entanto, isto não é necessariamente ruim, pois pode haver um outro programa com funcionalidade equivalente e qualidade superior que atraia o foco dos usuários. O modelo de desenvolvimento também favorece a possibilidade de bifurcação (*forking*) nos projetos (SOFTEX, 2005). Se existe um desacordo quanto ao rumo que o desenvolvimento deve tomar, sempre há a possibilidade de se iniciarem linhas paralelas de desenvolvimento aproveitando as linhas de código já escritas, enquanto o projeto original segue suas diretrizes. Ao final do seu artigo, já dizia Raymond:

Eu acredito que o futuro do software de código aberto irá pertencer gradativamente

---

21 Raymond (1997)

às pessoas que saibam como jogar o jogo Linus, pessoas que deixam para trás a catedral e abraçam o bazar. Isto não quer dizer que uma visão individual e brilhante não irá mais ter importância; ao contrário, eu acredito que o estado de arte do software de código aberto irá pertencer a pessoas que iniciem uma visão individual e brilhante, então amplificando-a através da construção efetiva de uma comunidade voluntária de interesse.<sup>22</sup>

---

22 Ibid.

## 3. PROPRIEDADE INTELECTUAL E CULTURA LIVRE

### 3.1 Propósito e breve histórico da propriedade intelectual

O que se convencionou denominar propriedade intelectual é na verdade um conjunto amplo de sistemas de proteção. Os elementos mais importantes deste conjunto são os segredos comerciais, as marcas registradas, as patentes e o direito autoral. A discussão que propomos neste trabalho envolve necessariamente a centralidade na questão do direito autoral. Como proposto no capítulo anterior, a concepção implícita de software que delineamos nos leva menos a uma análise sobre a questão das patentes do que a do direito autoral. Essa ressalva, no entanto, não desqualifica a pertinência dos argumentos relacionados. Como sistemas fundados por uma mesma idéia, ambos partilham do substrato único da propriedade intelectual.<sup>23</sup>

Para todos os efeitos, a propriedade intelectual é um caso específico da propriedade privada. Ela articula e garante o direito de uso e disposição de um bem, com a diferença que este, enquanto propriedade intelectual, é em princípio desprovido de materialidade. Essa especificidade elementar qualifica a propriedade intelectual com uma dimensão problemática, e sua natureza peculiar foi bem compreendida pelos seus teóricos. Thomas Jefferson, dentre outras coisas, um dos principais intelectuais da Independência Americana e um dos primeiros responsáveis pelo escritório de patentes dos Estados Unidos, em uma de suas cartas, enunciou uma das mais relevantes reflexões sobre a questão:

If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but at the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because

---

23 Despidendo-se momentaneamente da acuidade conceitual jurídica, não é incorreto afirmar que uma patente é uma forma de uso do direito de autoria. Uma patente é um direito conferido pelo Estado que dá ao seu titular a exclusividade da exploração de uma tecnologia (solução original de um problema técnico) como contrapartida pelo acesso do público ao conhecimento dos pontos essenciais do invento. Por outro lado, o direito autoral é o direito exclusivo de utilização, publicação ou reprodução de uma obra. Pode-se dizer que as patentes estão para as invenções assim como os direitos autorais estão para as criações artísticas. Essa sutil diferenciação implica em consequências práticas consideráveis: a legislação norte-americana, por exemplo, ao permitir o direito a patentes de software, abre a possibilidade que certas formas, bastante genéricas, de se efetuar um procedimento venham a ser (e de fato o são) alvo de uma solicitação de uso proprietário. O “*one click shopping*” da Amazon.com é um caso emblemático dessa situação. Nesses termos, alguns autores descrevem as patentes de software como “minas terrestres para programadores”. No Brasil, software é considerado direito autoral.

every other posses the whole of it. He who receives an ideia from me, receive instruction himself without lessening mine; as he who taper at mine, receives light without darkening me. That ideia should freely spread from one to another over the globe, for the moral and mutual instructions of man, and improvement of his condition , seems to have been peculiary an benevolently designed by nature, whom she made them, like fire, expansible over all space, whitout lessening density at any point, and like the air in with we breathe, move, and have our physical being, incapable of confinement or exclusive.<sup>24</sup>

Ortellado (2002) afirma que com esse argumento a intenção de Thomas Jefferson era desqualificar a justificativa da propriedade intelectual como direito natural. Essa primeira linha de pensamento, protagonizada pelos defensores mais radicais do recrudescimento dos sistemas de direito intelectual, argumenta que, a exemplo da propriedade de bens materiais, a propriedade intelectual seria um direito natural. Simon e Vieira (2007) elucidam:

A naturalidade de um direito, nessa visão, implica que ele não deve ser limitado em visão ao bem comum, por exemplo. Assim, a propriedade intelectual não teria caráter pragmático ou histórico, mas seria o simples reconhecimento de um direito absoluto do homem sobre sua produção intelectual.

E apontando outra corrente, prosseguem:

Outra linha de pensamento é a de que esses sistemas de proteção são socialmente construídos, e que portanto (reflitam ou não um direito adicional, intrínseco ao homem) devem ser limitados de forma a cumprir apenas a finalidade para a qual foram concebidos; eles seriam construções artificiais, que não refletem necessariamente a natureza humana, mas sim um cálculo em vista de determinados fins (por exemplo, a promoção das artes e das ciências). Assim, a finalidade e a eficácia de eventuais leis de propriedade intelectual deveria ser julgada tendo em vista o bem que as institui, ou seja, o bem da sociedade como um todo.

O “bem da sociedade” é portanto o critério que autoriza “esses sistemas de proteção [...] a cumprir apenas a finalidade para qual foram concebidos”. A lógica implícita assevera que as criações intelectuais, justamente por possuírem a peculiaridade de uma vez expressadas, serem assimiladas por todos que a recebem, devem ser especialmente protegidas, para que os criadores não fiquem desestimulados a criá-las e expressá-las. Aquele que cria a idéia deve ter o direito sobre ela, de forma que toda a vez que alguém a utilize ou a receba, ele tenha uma recompensa material (Ortellado, 2002). O artista deve receber os direitos autorais pela sua obra e o inventor, o direito pelo uso da patente. Assim, o incentivo consiste em oferecer ao inovador a concessão de monopólios restritos.

Este princípio foi incluído na Constituição dos Estados Unidos no fim do século 18. O

---

24 Simon (2000).

artigo I, seção 8, cláusula 8, estipula a seguinte responsabilidade ao congresso: “*promote the Progress of Science and useful Arts, by securing for limited Time to Authors and Inventors the exclusive Right to their respective Writings and Discoveries*”.<sup>25</sup>

A motivação básica nesse ambiente legal, para o estabelecimento da propriedade intelectual, é o desejo de incentivar a produção intelectual. Com o direito exclusivo às suas criações, oferece-se, em contrapartida, a justa recompensa aos autores pelo seu empenho e talento. A possibilidade de explorar comercialmente as suas idéias é o estímulo para que o criador produza ainda mais e a sociedade progrida em direção ao bem comum (Ortellado, 2002).

Posto a finalidade e o argumento do incentivo, e negando-se a tese do direito natural, a consideração do esquema de propriedade intelectual invariavelmente remete à dosagem adequada de sua implementação. Nas palavras de Ortellado (2002): “ a questão é até que ponto a introdução do direito de propriedade intelectual , ao invés de promover, termina por constranger o processo do saber, da cultura e da tecnologia. Se os critérios para se estabelecer a propriedade são rígidos e a duração do direito longa demais, então, pode-se dificultar o aproveitamento social da criação”. A balança idealizada contrapõe o direito do autor em ser remunerado e o interesse da sociedade em usufruir de suas criações ao menor tempo possível e com o menor custo. O monopólio garantido, suficientemente ruim para a transação de bens materiais, ganha novo significado em bens despidos de materialidade.

A característica mais saliente da idéia de propriedade é a possibilidade que ela oferece de excluir outras pessoas de terem acesso ao bem em questão. Isso vale claramente para a propriedade física, rival; essa exclusão é a razão pela qual a propriedade é fundamental para a nossa sociedade (capitalista e baseada no Direito):é ela que possibilita a comercialização de bens rivais e o cumprimento dos contratos que envolvem bens rivais. [...] a propriedade intelectual aplica-se a bens que não são rivais. No caso deles, a exclusão não é necessária para garantir o cumprimento de contratos; ela justifica-se, argumenta-se, por garantir a comercialização por meio da escassez, e com isso incentivar a produção de novos bens. [...] a escassez é claramente uma desvantagem trazida pela propriedade intelectual. Desvia para o uso privado bens que, de outra forma, teriam uso público; torna bens naturalmente abundantes em artificialmente rivais, e portanto escassos. A questão, então, passa a ser se as vantagens que ela eventualmente proporciona são maiores (ou simplesmente preferíveis) que as desvantagens que ela certamente acarreta (SIMON, VIEIRA, 2007).

A justa remuneração do autor, portanto, implica na introdução de um viés de escassez em um ambiente que, a princípio, desconhece esse plano. O desserviço que a exclusividade de uso e comercialização proporciona, argumenta-se, ainda assim se coloca como a maneira

---

25 Simon (2000).

menos excepcional de remuneração. Nesses termos, é desejável que os autores sejam remunerados, mesmo que, para tanto, uma prática não-desejável ocorra. Para que se consiga o que é bom, devemos nos submeter ao que é ruim<sup>26</sup>, e portanto, toda a questão remete a saber em que proporção exata essa submissão é necessária. Está é a preocupação prática que perpassa e tem acompanhado todas as discussões sobre a implementação das legislações de propriedade intelectual.

Na Inglaterra, a pioneira na conceituação formal e efetivação da propriedade intelectual por ocasião do “*Statute of Anne*” em 1709<sup>27</sup>, a primeira lei concedia ao criador o direito exclusivo sobre sua obra (primordialmente, a legislação tratava somente sobre o mercado editorial) por um prazo de catorze anos, prorrogáveis por mais catorze anos se o autor estivesse vivo quando o direito expirasse. A legislação americana inspirou-se na inglesa e em 1790, o período de catorze anos, renováveis por mais catorze, também passou a vigorar nos Estados Unidos. Em 1831, o congresso americano iniciou um processo de revisão das leis de direito autoral e o período inicial expandiu-se dos catorze anos iniciais para vinte e oito, permanecendo a possibilidade de renovação por mais catorze anos. Em 1909, novamente uma revisão das leis ampliou o direito de renovação de catorze para vinte e oito, totalizando uma possibilidade de cinquenta e seis anos de direito autoral sobre uma obra.

Após a segunda metade do século XX, com o incremento do poder da indústria cultural<sup>28</sup>, uma nova rodada de “atualizações” se fez presente. Em 1955, o congresso norte-americano requisitou um estudo ao escritório de patentes a fim de subsidiar novas reformulações das leis de direito autoral vigentes. O parecer final do relatório recomendava uma ampliação do período de renovação de vinte e oito para quarenta e oito anos. A indústria cultural (principalmente as editoras), no entanto, propunha um período que contemplasse a vida do autor mais cinquenta anos após sua morte. O eminente término de algumas preciosas licenças por parte da indústria, em conjunto com as reiteradas objeções do Departamento de Justiça, levaram a disputa a um impasse, de tal forma que somente após nove adiamentos extraordinários, de 1965 a 1976, o Congresso finalmente aprovou uma nova lei de direitos

---

26 “Para que consiga o que é bom, devemos nos submeter ao que é ruim” é uma sentença, proferida pelo historiador britânico Thomas Babington Macauley, que reflete com acuidade a natureza do argumento proposto pela segunda linha de pensamento, conforme supracitado. Essa referência é encontra-se em Ortellado (2002).

27 Simon (2000).

28 Indústria Cultural é o nome genérico que se dá ao conjunto de empresas e instituições cuja principal atividade econômica é a produção de cultura, com fins lucrativos e mercantis. No sistema de produção cultural encaixam-se a televisão, o rádio, jornais, revistas e entretenimento em geral; que são elaboradas de forma a aumentar o consumo, moldar hábitos, educar, informar, podendo pretender ainda, em alguns casos, ter a capacidade de atingir a sociedade como um todo.

autorais. A nova reformulação atribuía um período de vigência do direito autoral por toda a vida do autor mais cinquenta anos e, para trabalhos encomendados por empresas, um período de setenta e cinco anos após a publicação ou cem anos após a criação, o que fosse mais curto.

No começo de 1990, novamente por força da expiração de um série de licenças, um novo movimento capitaneado pela indústria cultural começou a pressionar por um novo incremento dos prazos de vigência do direito autoral. O resultado foi a ampliação, em 1998, dos direitos autorais após a morte do autor de cinquenta para setenta anos, caso o direito fosse propriedade de uma pessoa, e a ampliação de setenta e cinco para noventa e cinco anos caso o direito fosse propriedade de uma empresa.<sup>29</sup>

Após quase três séculos, a prática da política de direitos autorais demonstrou que existem algumas áreas de dificuldades associadas a ela. Por um lado, a tecnologia de fazer cópias evoluiu constantemente e substancialmente com o tempo, dificultando a imposição da lei e chegando a torná-la inefetiva. Como já colocado, a prerrogativa do monopólio comercial, e portanto o controle do uso e disseminação de determinada propriedade intelectual, é um dos elementos fundadores do direito autoral. A possibilidade, provida pela tecnologia, de se banalizar o ato da cópia sempre constituiu um subterfúgio a esse controle.

Desde que os direitos de propriedade intelectual foram instaurados, houve um resistência aberta à sua aplicação no setor privado e comunitário. A enorme dificuldade de fiscalização fez com que essa desobediência civil tivesse um caráter passivo, que não se engajava nas contestações das leis de propriedade intelectual, mas simplesmente as ignorava. As pessoas sabiam que os direitos existiam e deveriam ser respeitados e simplesmente passavam por cima dele porque achavam que eram absurdos. Evidentemente não estou me referindo à pirataria comercial que era, sem exagero, apenas crime. A indústria pirata reconhecia a legislação vigente e fugia dela de forma clandestina sem contestá-la. Aliás, todo industrial pirata não podia aspirar a coisa maior do que transformar sua indústria legal e passar assim a utilizar os direitos autorais a seu favor (ORTELLADO, 2000).

Por outro lado, a cópia inegavelmente desempenha um papel extremamente positivo e altamente desejável para o progresso das sociedades em geral e para a preservação e incremento das suas culturas.

Mas coisa muito diferente eram os usuários que reproduziam a obra para fins não comerciais [...] Quando aparelhos de reprodução se popularizaram (o mimeógrafo, a fita cassete, a copiadora e em seguida a reprodução digital por computador), as pessoas automaticamente começaram a reproduzir livros, canções, fotos e vídeos, para si e seus amigos, sem pagar os devidos direitos, assim como, antes, já encenavam peças nas escolas e nos bairros e cantavam e tocavam canções para os amigos e para a comunidade também sem pagar os direitos. Por mais que a campanha “cívica” promovida pela indústria e pelo governos lembrassem a todos a

---

29 Esse breve histórico de mudanças nas leis autorais foi extraído de Ortellado (2000) e Lessig (2004).

importância de 'pagar os direitos', as pessoas desconfiavam, frequentemente de forma intuitiva, que aquele pagamento não fazia sentido pois quem apenas usufruía desse bem coletivo que é a cultura humana não podia estar roubando nada de ninguém. Como Benjamim Franklin havia escrito na sua autobiografia, na produção de cultura (e do saber e da tecnologia), nada pode ser feito sem que se antes tenha aprendido com a imensa comunidade dos outros produtores contemporâneos e dos que nos precedem. E da mesma forma que usufruímos e aprendemos gratuitamente com todos eles, de maneira tão ampla que sequer podemos nomeá-los individualmente, devemos disponibilizar nossas contribuições para a formação das novas gerações (ORTELLADO, 2002).

O recrudescimento da proteção à propriedade intelectual não é, portanto, necessariamente um escolha do tipo “ganha-ganha”. Ela é, como a maioria das decisões referentes a políticas públicas, um *trade-off* : propriedade intelectual mais ampla implica numa sinalização mais contundente aos incentivos individuais da produção intelectual ao mesmo tempo que dificulta a apropriação e criação destas mesmas obras. Outro aspecto importante a realçar é que esses princípios não são universais, e embora amplamente aceitos internacionalmente, há uma considerável variação nos detalhes, refletindo a dificuldade de se chegar a um bom termo. De fato, esses mecanismos e até mesmos os princípios envolvidos mudam de país para país.

### **3.2 Copyleft e commons intelectual**

Sob o ponto de vista da propriedade intelectual, certamente o copyleft é uma das maiores, senão a maior, inovações nessa área desde a concepção do copyright. Como já explicado, o copyleft - na premissa de que o autor tem, em boa parte, o direito de decidir sobre o destino e o uso (especificamente sobre o uso e disposição em obras derivadas) do bem intelectual que produz - atua sobre o direito autoral invertendo o funcionamento usual da propriedade intelectual. O criador autoriza que aquela obra seja usada, derivada e redistribuída irrestritamente, desde que sob as mesmas condições. Ou seja, as obras redistribuídas e/ou derivadas também devem ser redistribuídas e/ou derivadas irrestritamente - sempre sob os termos da mesma licença. É dessa forma que se operacionaliza as premissas (as quatro liberdades) do software livre.

Note-se que o copyleft também é um mecanismo de exclusão. A obra que adentre esta modalidade de licença inaugura uma cadeia de derivações que sempre terá as mesmas condições de uso.<sup>30</sup> Nos termos de um conjunto crescente de bens culturais livres, o copyleft

---

30 Trata-se, é certo, de uma exclusão consentida: está nas mãos de cada um a possibilidade de aderir.



também pode ser entendido como um ferramenta de aglutinação e construção. Para entender essa perspectiva, é bastante útil, talvez fundamental, entender a idéia de commons<sup>31</sup>.

Podemos definir commons como um conjunto de recursos utilizados em comum por uma determinada comunidade. Todos os membros dessa comunidade podem utilizá-los, de forma transparente, sem necessidade de permissão de acesso. No entanto, em commons bem sucedidos usualmente há regras e convenções que delimitam o uso responsável para que os recursos não se extingam ou se deterioreem. O exemplo clássico de commons remete a áreas de pasto, florestas e outros recursos naturais utilizados de forma cooperativa, sem que qualquer indivíduo possa clamar propriedade sobre os mesmos. Via de regra, todo recurso ambiental compartilhado ou bens essenciais à vida na cidade, como ruas, parques e pontes, são commons.

A natureza material desses commons implica num claro limite: a sua esgotabilidade. Se um rebanho se alimenta de uma determinada área de pastos, será necessário que as plantas voltem a crescer para que um novo rebanho venha e se alimente; uma área litorânea não produzirá pesca o suficiente caso haja um excesso de pescadores. Dessa forma, o uso simultâneo desses recursos obedece a limites claros. Esse perfil se traduz numa característica de rivalidade<sup>32</sup> no uso desses bens. Por esse motivo, commons materiais são chamados de extinguíveis, competitivos ou rivais. Simon e Vieira (2007) argumentam que, embora a análise clássica alegue a sua inevitável tendência à extinção (“pois a liberdade que permitem conduziria a uso irresponsável e excessivo”), uma nova compreensão desse fenômeno qualifica a viabilidade dessas iniciativas, desde que determinadas condições (principalmente ligadas à regulamentação do uso possível) sejam observadas.

Recentemente, passou-se também a conceber commons baseados em bens não-rivais; entre eles está o que vem sendo chamado de commons intelectual, justamente por compor-se de bens intelectuais. Como já marginalmente colocado na seção anterior, os bens intelectuais são não-rivais porque não é possível provocar a escassez de um determinado conhecimento pelo seu uso excessivo. Eles são abundantes: o uso de um indivíduo não interfere no uso de outro. Rememoramos Thomas Jefferson: “If nature has made any one thing less susceptible than all others of exclusive property, it is the action of thinking power called an idea”.

Há, entretanto, um outro dilema que afeta tanto bens rivais e não-rivais. Trata-se da

---

31 Uma tradução razoável para o termo commons seria “coisa pública”.

32 Rivalidade é a situação em que o consumo de um bem por uma pessoa impede o consumo do mesmo bem por outra pessoa. Atentamos ao fato que a característica de rivalidade não é condição suficiente para que o bem seja suprido por mercados privados. Nesse caso é necessário a existência de exclusibilidade, definido como a possibilidade de se excluir um consumidor pela capacidade de pagar pelo consumo de um bem ou serviço.

expressão, no âmbito dos commons, do mesmo problema de incentivo tão comum à discussão da propriedade intelectual. O problema do provisionamento traduz-se na questão de como garantir que haverá incentivo suficiente para que os bens sejam produzidos.

Nesse quesito, afirmam Simon e Vieira (2007), a experiência do software livre oferece uma solução possível:

Ele é anti-rival [o software livre]. O meu uso do software livre não só não rivaliza com o uso de outra pessoa; pelo contrário, cada pessoa a mais que o usa auxilia os demais usuários - seja contribuindo de volta ao commons desenvolvendo o software (isto é, programando), identificando erros, ou simplesmente indicando características novas que poderiam ser implementadas; e torna os programas mais valiosos pois a adoção em larga escala do software aumenta a comunicabilidade e compatibilidade entre os membros da comunidade que se forma ao redor desse software: se todos utilizam um mesmo editor de textos, por exemplo, podem trocar livremente arquivos entre si. Trata-se de um efeito de rede; o valor de uma rede cresce mais do que cresce o seu número de nós<sup>33</sup> [...]. Com efeito, a característica do commons é justamente permitir o uso coletivo e indiscriminado de seus bens. Além disso, no caso do commons intelectual não existem restrições quanto ao número de usuários, já que os bens intelectuais são naturalmente abundantes.

O copyleft, enquanto um dos fundamentos do modelo de desenvolvimento de software livre, constitui um mecanismo bastante adequado à viabilização de commons intelectual. Toda a produção de bem intelectual - seja produção científica ou cultural - tal como a conhecemos tem entre seus fundamentos implícitos diversos tipos de recursos a informação já existente. Assim, no processo de geração de cultura e conhecimento, os bens intelectuais são tanto um produto final como insumo. O software livre caracteriza de maneira clara este aspecto: o acesso ao código fonte permite múltiplas maneiras de se escrever um programa.

Hoje, a circunstância social, econômica e tecnológica possibilita a implantação de commons intelectuais com uma facilidade inédita na história. As informações digitalizadas, cujo custo de armazenamento e distribuição é praticamente nulo, permitiram um salto significativo na eficiência da distribuição de bens intelectuais. Com a Internet, o custo para enviar e receber a cópia de um arquivo digital se tornou desprezível. A qualidade dessas cópias também evoluiu de tal maneira que se tem uma reprodução muito mais fiel da composição original. Naturalmente, esse nunca foi um problema na esfera dos softwares, no entanto, em outros tipos de produção cultural, até científica, esse detalhe sempre influenciou o seu grau de disseminação. Com esses elementos, consegue-se uma melhor implementação física dos commons não-rivais. Simon (2000) pondera:

---

33 “O valor de uma rede cresce mais do que cresce o seu número de nós” é uma enunciação menos precisa da famosa lei de Metcalfe, onde o valor da rede é o quadrado do seu número de nós.

A cópia é uma operação extremamente importante e poderosa no mundo digital. A natureza preserva a vida há um bilhão de anos através da operação da cópia digital! Usando a cópia de símbolos como operação principal uma máquina de Turing é capaz de implementar qualquer procedimento algorítmico! Esta mesma idéia foi aproveitada pelo computador da IBM 1620, no início dos anos 60. Neste computador até mesmo a soma e a multiplicação era realizados pelo uso engenhoso da cópia da informação. Aristóteles já havia enunciado que o homem é o maior mímico de todos os animais. De fato a cópia é um mecanismo essencial para nossa existência social, para nossa aprendizagem e para nossa evolução. [...] a cópia no contexto da Internet pode levar a resultados significativos e até mesmo surpreendentes. Será que restringir a cópia ainda é o melhor mecanismo de incentivar a produção intelectual ?

Com a virtual banalização dos custos de disseminação de um conhecimento, e com as possibilidades que a idéia de commons intelectual enseja, especialmente quando pensamos em experiências concretas como o software livre, há de se questionar se a imagem tradicional da balança que opõe estímulo material ao criador e interesse social em usufruir a obra ou invenção não pendeu para este último. Talvez em situações (que pode encaminhar para ser a única situação) onde “o valor da uma rede cresce mais do que cresce o seu número de nós” seja do interesse individual do criador que o maior número possível de indivíduos tenha acesso a sua obra.

O software livre, entendido como um commons intelectual, serviu para deixar claro o *trade-off* intrínseco à utilização da forma usual de propriedade intelectual. De fato, o copyleft foi instrumental na construção dessa comunidade forte e coerente que conseguiu mostrar a enorme força de um commons dinâmico e rico em conteúdos.

### **3.3 Creative Commons e cultura livre**

Considerando o commons intelectual como ponto de partida, se tomarmos o espectro de casos concretos, no qual o critério é o grau com que eles facilitam a reprodução e ampliação do commons, teríamos entre o movimento software livre e o seu derivado, a OSI (esse em menor sintonia com o critério estabelecido do que aquele), uma terceira iniciativa que, embora nasça profundamente inspirada pelo conceito de copyleft, se traduz no transbordamento deste para fora do mundo dos softwares assim como sua potencialização.

O projeto Creative Commons é a última novidade na trilha inaugurada pelo copyleft. Lançado oficialmente em 2001, embora a primeira proposta de licença só tenha sido publicada em dezembro de 2002, trata-se de um esforço para “disponibiliza[r] opções flexíveis de licenças que garantam proteção e liberdade para artistas e autores”. Argumenta-se

que “partindo da idéia de 'todos os direitos reservados' do direito autoral tradicional nós a recriamos para transformá-la em 'alguns direitos reservados’”.<sup>34</sup> O mote “opções flexíveis de licenças” fundamenta a sua colocação em meio ao software livre (tipificado pela licença copyleft padrão: a GNU GPL) e a OSI (em cuja referência apresentamos a licença BSD). As licenças Creative Commons foram pensadas como uma ferramenta de uso amplo, para quaisquer produtores de conteúdo (não apenas de software) que se interessem em maneiras alternativas de restringir seus direitos autorais. Elas são altamente configuráveis; a única característica fixa em todas as licenças é a exigência de atribuição (nos casos de citação, redistribuição ou derivação). Há as opções de regular (exigindo, permitindo ou proibindo) o uso não-comercial, a derivação, a redistribuição livre etc.; há ainda licenças que cobrem casos particulares, como as que permitem determinados usos apenas em países pobres. O fato de que, em princípio, seja possível usar essas licenças para se aplicar o direito autoral usual, caracteriza o Creative Commons como a mais extrema “liberdade”<sup>35</sup> no âmbito da propriedade intelectual.

Embora vise um espectro mais amplo de usos do que o software livre, o projeto Creative Commons fortalece o commons intelectual, pois todas as suas licenças permitem no mínimo a redistribuição não-comercial do produto. Ele tem o potencial de agregar mais produtores para a causa da flexibilização dos direitos autorais, tanto pela gama de opções que oferece, como por não restringir-se a software, e portanto dirigir-se a um universo maior de produtores. O projeto é muito bem sucedido: após cinco anos de vida, existem hoje mais de 140 milhões de páginas na web usando as licenças CC. Isto certamente mostra que a idéia é bem recebida por amplas camadas da sociedade (SIMON, VIEIRA, 2007).

Licenças mais flexíveis ensejam a participação de uma comunidade maior, mas potencialmente diminuem as possibilidades de contribuição ao commons intelectual. Por outro lado, modalidades que mais se aproximam do copyleft contribuem mais ao commons a cada bem intelectual licenciado, no entanto tendem a atrair menos interessados já que reservam menos direitos ao autor.

Essa peculiar posição que as licenças Creative Commons ocupam, a meio caminho entre o direito autoral tradicional e as experiências alternativas, reforça a posição que seu fundador, Lawrence Lessig, advoga. O tema da cultura livre, cuja esquematização mais clara encontra-se no livro homônimo *Free Culture*<sup>36</sup> de Lessig, articula algumas preocupações que

---

34 <<http://www.creativecommons.org.br/>>

35 A “liberdade” para se fazer o que quiser, inclusive cercear o uso, em contraposição a “liberdade” idealizada pela FSF, constitui, como rapidamente expusemos no primeiro capítulo, uma modalidade negativa. É certamente uma maneira idiossincrática de elucidar o termo.

36 A similaridade com *free software* se explica: “Como nos argumentos de Stallman por um software livre, um

se imbricam com a temática do software livre. Na linha da discussão sobre propriedade intelectual, qualificamos estas idéias menos como uma tese do que um conjunto de propostas que buscam adequar novas modalidades mais flexíveis de direito autoral às práticas de compartilhamento e criação no âmbito da Internet.

A identificação da Internet como uma força central nas possibilidades de uso compartilhado e criativo de bens intelectuais (*creatives* na terminologia de Lessig) é proeminente na noção de cultura livre. De fato, Lessig, ao recordar como a indústria radiofônica, na primeira metade do século XX, combateu e quase levou ao insucesso o padrão FM - em termos técnicos infinitamente superior ao padrão AM, que até então era dominante - demonstra como os “poderes econômicos estabelecidos” repetem o mesmo perfil de ação ao impedir que as leis de copyright se adequem às novas tecnologias.

A questão, argumenta Lessig, é dimensionar os efeitos e possibilidades que as novas tecnologias ensejam, garantindo que estas novas formas não sejam cerceadas por uma estrutura de direitos autorais inflexível, simbiótica com modelos de atuação<sup>37</sup> já consolidados.

Essa tradição é o modo como nossa cultura foi construída. [...] Uma cultura livre apóia e protege os criadores e inovadores. Ela faz isso diretamente garantindo direitos sobre a propriedade intelectual. Mas ela o faz também indiretamente limitando o alcance de tais direitos, garantindo que os futuros criadores e inovadores mantenham-se *o mais livre* possível dos controles do passado. Uma cultura livre não é uma cultura sem propriedade, da mesma forma que um mercado livre não é um mercado aonde tudo é liberado. O oposto de uma cultura livre é uma cultura de permissão – uma cultura na qual os criadores podem criar apenas com a permissão dos poderosos ou dos criadores do passado (LESSIG, 2004).

A liberdade, portanto, da “cultural livre” remete a uma proposta de fino ajuste dos incentivos de direitos autorais com o inegável ambiente de apropriação social permitido pelas novas tecnologias. Esse ajuste exacerba a prerrogativa do autor escolher qual fim dará ao seu trabalho, dependendo, no caso, exclusivamente de suas motivações.

O modelo de software livre, conforme exposto no primeiro capítulo, atesta que a prática de um integral retorno das contribuições individuais ao commons constituído possibilita avanços de qualidade indiscutível. Por outro lado, contribuições voluntárias, como as obras sobre licenças do tipo Creative commons, no caso mais amplo da cultura, e softwares registrados com as licenças OSI, parecem também apontar que nesses casos existe

---

argumento por uma cultura livre esbarra em uma confusão que é difícil de se evitar, e ainda mais difícil de entender. Uma cultura livre não é uma cultura sem propriedade; não é uma cultura aonde os artistas não são pagos. Uma cultura sem propriedades, onde os artistas não são pagos, é uma anarquia, não liberdade” (LESSIG, 2004).

37 Nos termos da indústria cultural, o que chamamos de modelos de atuação já consolidados são os grandes conglomerados econômicos que atuam no mercado de mídia.

uma situação superior a produzida por esquemas de propriedade intelectual mais rígidos.

Simon e Vieira, no entanto, parecem discordar. Qualquer aumento da dificuldade de um bem intelectual voltar ao commons é contraproducente, afirmam os autores. O fato de os bens intelectuais serem produto e insumo ao mesmo tempo significa que com a redução do commons, eles tornam-se menos plurais e diversificados, além de mais caros. Essas condições, supõe-se, restringem o ciclo evolutivo dos bens intelectuais disponíveis

Ainda que a propriedade intelectual (de qualquer forma, flexível ou não) somente afaste temporariamente<sup>38</sup> o bem intelectual do commons, permanecem três problemas fundamentais.

Em primeiro lugar, ele subordina o commons à propriedade intelectual, na medida em que afirma que algo só chega ao commons depois de passar pela propriedade intelectual. É mais razoável, no entanto, dizer o contrário: que os bens intelectuais deixam o commons – se a sociedade assim determina – são temporariamente sujeito a propriedade intelectual; e depois disso retornam ao commons. Essa maneira de ver a cadeia de produção da informação é simplesmente o reconhecimento de que a produção de toda informação toma como *input* outra informação. É também o reconhecimento de que os bens intelectuais “nascem” no commons, e não sob propriedade intelectual [...] Em segundo lugar, o argumento ignora o fato de o commons intelectual é uma estrutura complexa, viva como a cultura humana, ele não existe de forma estática, mas como processo, em constante reatualização e mutação. [...] Em terceiro lugar, o argumento não leva em conta a perspectiva histórica do longo prazo, e não oferece evidências contrárias a essa perspectiva. Olhando a história humana, constata-se que, durante a maior parte do tempo, nossa cultura prosperou sob regimes frouxos (ou inexistentes) de “propriedade intelectual”, que privilegiavam um commons intelectual amplo. Por outro lado, ainda não há evidências sólidas de que a inversão dessa balança irá manter (muito menos de que irá melhorar) a vitalidade da cultura e da ciência (SIMON, VIEIRA, 2007).

Como um exercício normativo, os diferentes caminhos apresentados possuem argumentos suficientemente robustos. Como uma política pública, o sistema de propriedade intelectual é cabível de ser alterado se o devido entendimento estiver presente.

---

38 Se bem que em prazos tão longos como os propostos pela legislação americana, esse “temporariamente” foge a qualquer defesa razoável.

## 4. CONSIDERAÇÕES FINAIS

Como procuramos ressaltar, o software, mais que um simples artefato tecnológico, possui certas características que permitem uma definição um tanto diferente do que normalmente se pensaria. A sua interpretação como uma linguagem o coloca no mesmo grupo das produções culturais, e talvez até científicas.

Essa natureza peculiar, junto com a inexistência de um estímulo econômico mais relevante fizeram com que, no princípio da informática, a prática usual entre a comunidade de programadores e usuários fosse a livre distribuição e alteração dos programas. Com a reversão dessa tendência, em 1980, um movimento capitaneado por Richard Stallman surgiu questionando a imposição de licenças proprietárias ao ambiente que antes era caracterizado pelas liberdades de uso, distribuição e alteração. Essas mesmas liberdades vieram a se tornar a proposta central de um software livre. Como vimos, há diferenças entre um software de código aberto e um software dito livre. Mais do que um exercício de retórica, elas se revelam especialmente importantes na definição da licença de uso.

O primeiro pilar do software livre é justamente a construção de modelos de licenciamento. A efetivação das liberdades preconizadas pela FSF se viu viabilizada com a criação de um mecanismo legal de proteção ao livre acesso do código fonte. O copyleft, ao exigir legalmente que qualquer trabalho derivado tenha as mesmas modalidades de uso, disponibiliza uma poderosa ferramenta de confronto ao software de código fechado. Essa característica se tornou o eixo de uma dinâmica que permitiria ao movimento pelo software livre desenvolver o segundo pilar de sua proposta.

O modelo de desenvolvimento de software livre é hoje um dos mais forte argumentos à utilização disseminada do softwares com código fonte acessível. A liberdade de uso, modificação e redistribuição, em conjunto com a trajetória ascendente da Internet permitiu que interesses e vontades dispersos na rede formassem movimentadas comunidades de desenvolvimento. Para a surpresa de todos, esses coletivos não só fundaram um novo e eficiente modelo de criação e desenvolvimento de programas, mas como o fizeram de uma maneira colaborativa, horizontal, constituída de motivações difíceis de entender sob a lógica do preço, do mercado. A proposta do software livre, entretanto, não é excludente com a utilização comercial dessas iniciativas. Ela apenas incentiva a mudança de foco da comercialização de licenças de uso para outros serviços.

Como procuramos demonstrar, ambos esses elementos, o copyleft (modelo de licenciamento) e o modelo de desenvolvimento, notadamente o primeiro, articulam necessariamente o tema do software livre com a discussão do limite da propriedade intelectual.

Sem dúvida o software livre é a primeira grande iniciativa social a demonstrar o grande e incompreendido poder das novas tecnologias de comunicação e informação, por um lado, e uma resposta prática ao que foi por muito tempo considerado a razão de ser dos sistemas de proteção a propriedade intelectual, por outro.

Apontamos como o direito autoral, assim que é refutada a tese do direito natural, foi fundado sobre o entendimento de que a imaterialidade dos bens intelectuais, e portanto a dificuldade de apropriação que um criador teria com os frutos do seu trabalho, seria altamente prejudicial aos incentivos para à produção intelectual. A implementação dos direitos de propriedade intelectual, portanto, teria uma finalidade.

Essa finalidade seria o critério para definir em que medida os autores deveriam ter o direito de uso sobre suas obras, sendo o monopólio comercial a principal recompensa pelos seus esforços criativos. Essas restrições de uso seriam um desserviço menor em comparação a grande finalidade proposta. A questão resumir-se-ia a definir o ajuste entre esses dois lados. A história, no entanto, demonstrou como esse princípio original se deturpou ao longo de quase três séculos de existência. Como mostramos, a duração dos direitos autorais expandiu-se de, no limite, vinte e oito anos em 1709 com a primeira lei, para a atual duração <sup>39</sup> que pode muito bem extrapolar os cem anos.

As novas tecnologias, entre elas a de cópia, desenvolveram-se de tal maneira que a possibilidade de disseminação social dos bens intelectuais se tornou virtualmente sem custos. Essa possibilidade atesta que os esquemas usuais de propriedade intelectual introduzem um elemento de escassez onde em princípio ele não existe. Evidenciamos que esse aspecto enfatiza a característica de uma política pública, âmbito que usualmente não é associado à propriedade intelectual. Como uma política pública, os direitos autorais impõem um *trade-off*.

Com o conceito de commons, procuramos qualificar a idéia de copyleft como possivelmente a primeira inovação na área dos direitos de propriedade intelectual desde a sua formulação original. A existência de commons intelectuais, composto por bens não-rivais, subsidia uma análise preliminar sobre as possibilidades que outros tipos de proteção autoral venham a desempenhar papéis preponderantes.

---

<sup>39</sup> Tomamos como exemplo a legislação norte-americana, entretanto essa tendência foi e é perceptível em todo o mundo.



Apontamos como a inclusão de uma nova iniciativa, a Creative Commons, dentre os esforços do software livre e do software de código aberto extrapola a prática de licenças de uso mais flexíveis, do campo restrito dos software, para o campo da produção cultural como um todo.

Essa prática, que a literatura preliminar denomina “cultura livre”, enfatiza a configurabilidade das licenças como uma boa solução aos impasses que o novo âmbito social, tecnológico e econômico impõe à propriedade intelectual.

## REFERÊNCIAS BIBLIOGRÁFICAS

BORGES, Jussara; JAMBEIRO, Othon; SANTOS, João T. **Notas sobre a economia política do software: o caso de Salvador.** Disponível em <[www.rp-bahia.com.br/biblioteca/pdf/JoaoTiagoJesusSantos.pdf](http://www.rp-bahia.com.br/biblioteca/pdf/JoaoTiagoJesusSantos.pdf)>. Acesso em março de 2008.

BRETON, P. **História da Informática.** Tradução: Elcio Fernandes. São Paulo: Editora Universidade Estadual Paulista, 1991. p.185 -204.

DIPOLD, Rafael Dragheti. **Potencialidade Econômica do Software Livre.** Toledo, 2005. Monografia (Bacharelado em Ciências Econômicas) – Centro de Ciências Sociais, Universidade Estadual do Oeste do Paraná – *Campus* de Toledo, 2005. Disponível em: <<http://www.sigha.com.br/dipold/monografia.pdf>>. Acesso em: março de 2008.

FERNANDES, Jorge Henrique Cabral. **Qual a prática do desenvolvimento de software.** *Cienc. Cult.*, Abr 2003, vol.55, n.2, p.29-33. Disponível em <[http://cienciaecultura.bvs.br/scielo.php?script=sci\\_arttext&pid=S0009-67252003000200021&lng=pt&nrm=iso](http://cienciaecultura.bvs.br/scielo.php?script=sci_arttext&pid=S0009-67252003000200021&lng=pt&nrm=iso)>. Acesso em março de 2008.

HEXSEL, Roberto A. **Propostas de Ações do Governo para Incentivar o Uso de Software Livre.** Relatório Técnico do Departamento de Informática da UFPR. Curitiba, Paraná, 2003. Disponível em <[http://www.inf.ufpr.br/info/techrep/RT\\_DEINF004\\_2002.pdf](http://www.inf.ufpr.br/info/techrep/RT_DEINF004_2002.pdf)>. Acesso em março de 2008.

LESSIG, Lawrence. **Cultura Livre.** Disponível em <[www.rau-tu.unicamp.br/nou-rau/softwarelivre/document/?view=144](http://www.rau-tu.unicamp.br/nou-rau/softwarelivre/document/?view=144)>. Acesso em março de 2008.

NETO, Antônio; OLIVEIRA, Thiago. **Os limites da propriedade intelectual na fronteira do ciberespaço: uma análise do Software Livre a partir da Economia Política.** In: SIMPÓSIO INTERNACIONAL DE PROPRIEDADE INTELECTUAL, INFORMAÇÃO E ÉTICA. Florianópolis. 2003. Disponível em: <<http://www.ciberetica.org.br/trabalhos/anais/33-63-c3-1.pdf>>. Acesso em março de 2008.

ORTELLADO, Pablo. **Porque somos contra a propriedade intelectual?.** 2002. Disponível em: <<http://www.midiaindependente.org/pt/blue/2002/06/29908.shtml>>. Acesso em março de 2008.

RAYMAND, Eric S. **A Catedral e o Bazar.** 1997. Disponível em <<http://www.geocities.com/CollegePark/Union/3590/pt-cathedral-bazaar.html>>. Acesso em março de 2008.

SALEH, Amir M. **Adoção de Tecnologia: Um estudo sobre o uso de software livre nas empresas**. São Paulo, 2004. 123 f. Dissertação (Mestrado em Ciências Econômicas) – Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, 2004.

SIMON, Imre. A propriedade intelectual na era da Internet. **DataGramZero**, n. 1, 2000. Disponível em <<http://www.ime.usp.br/~is/papir/direitos/direitos-dgz.pdf>>. Acesso em março de 2008.

SIMON, Imre; VIEIRA, Miguel S. **A propriedade Intelectual diante da Emergência da Produção Social**. Disponível em <<http://www.ime.usp.br/~is/papir/pips-19mar07.pdf>>. Acesso em março de 2008.

SOFTEX, **O impacto do software livre e de código aberto na indústria de software do Brasil**. Softex, 2005, Disponível em: <<http://www.softex.br/media/pesquisa-swl.pdf>>. Acesso em março de 2008.

STALLMAN, Richard. **Linux e o Sistema GNU**. 1999. Disponível em <<http://www.geocities.com/CollegePark/Union/3590/linuxgnu.html>>. Acesso em março de 2008.

\_\_\_\_\_. **Projeto Gnu**. CIPSGA – Comitê de Incentivo a Produção do Software Gratuito e Alternativo. Disponível em: <<http://www.cipsga.org.br>>. Acesso em março de 2008.

\_\_\_\_\_. **Porque o software deveria ser livre**. 2001. Disponível em <<http://www.gnu.org/philosophy/shouldbefree.pt.html>>. Acesso em março de 2008.