

---

# POR DENTRO DA MENTE DE RICHARD STALLMAN: UMA ANÁLISE DO MOVIMENTO DO SOFTWARE LIVRE UTILIZANDO *CRITICAL SYSTEM HEURISTICS* (CSH)

F. G. CINTRA<sup>1</sup> e M. F. CAVALCANTI BANDOS<sup>2</sup>  
Sessão Temática A:

Teorias, conceitos e metodologias sistêmicas

[http://www.pucpcaldas.br/graduacao/administracao/revista/artigos/esp1\\_8cbs/artigos\\_8cbs\\_2012.html](http://www.pucpcaldas.br/graduacao/administracao/revista/artigos/esp1_8cbs/artigos_8cbs_2012.html)

---

## RESUMO

Palavras-chave:

---

### 1 Introdução

O presente artigo tem por objetivo analisar o software livre e o movimento dele originado à luz da metodologia sistêmica CSH (Critical Systems Heuristics), como proposta em Ulrich (1983), visando ampliar a compreensão do tema e de seus desdobramentos em diversas áreas da vida em sociedade. Para tanto, empenhou-se uma pesquisa exploratória (GIL, 1999) em que se levantaram dados secundários, nomeadamente a obra de Stallman (2010), idealizador do movimento em questão, a qual se emprega como corpus da análise.

Para fins de contextualização, efetuou-se uma releitura da história do movimento do software livre,

---

<sup>1</sup> Mestrando do Programa de Pós-Graduação em Desenvolvimento Regional do Centro Universitário de Franca – Uni-FACEF, da linha de Desenvolvimento Social e Políticas Públicas. Bacharel em Ciência da Computação com ênfase em Análise de Sistemas pela Universidade de Franca (1997) e em Direito pela Faculdade de Ciências Humanas e Sociais de Franca, da Universidade Estadual Paulista “Júlio de Mesquita Filho” – UNESP (2008). Responsável pelo Departamento de Informática do Uni-FACEF. E-mail: goncin@gmail.com.

<sup>2</sup> Doutora em Administração pela Universidade de São Paulo – USP (2008). Mestre em Administração pela USP (2003). Bacharel em Administração pela USP (1997) e em Direito pela Universidade de Ribeirão Preto – UNAERP (1998). Pró-Reitora de Extensão Comunitária e Desenvolvimento do Centro Universitário de Franca – Uni-FACEF. Docente do Programa de Pós-Graduação em Desenvolvimento Regional do Uni-FACEF. E-mail: melissa@facef.br.

pontuando os fatos e atos que engendraram seu surgimento e as inovações e contribuições que introduziu em diversos campos.

A análise sistêmica propriamente dita foi feita valendo-se de excertos de Stallman (2010) para responder às questões sugeridas pelo CSH. Obteve-se, como produto, um panorama abrangente do sistema estudado, não olvidados os elementos que o constituem e a relação de forças entre eles. Das considerações que despontam do resultado, é saliente a constatação de que o movimento do software livre ainda não atingiu o estado ideal imaginado por Stallman, embora tenha contribuído decisivamente para a mudança das relações de usuários e empresas de software que respeitam a este.

## **2 O movimento do software livre**

### **2.1 Revisitando a história**

O conjunto de fatos ocorridos nas origens do software livre é bastante conhecido, mas cumpre recontá-los para neles identificar seus elementos e as relações que, juntamente com o movimento e a comunidade dele originários, constituem o sistema de estudo.

Software livre, enquanto conceito, só veio a aparecer no início da década de 1980. Sem embargo, sua procedência remota pode ser rastreada até algumas décadas antes.

González-Barahona, Seoane Pascual e Robles (2009, p. 18-19) relatam que, nos anos 1960, o cenário de tecnologia da informação era dominado por computadores de grande porte (os mainframes), e a IBM dominava o mercado. Nessa época, ao se adquirir um computador (o hardware), o software era fornecido em conjunto. Enquanto vigorava o contrato de manutenção, o acesso ao catálogo de software do fabricante era franqueado ao cliente. Do ponto de vista comercial, a concepção de programas como um produto à parte era algo raro. Muito mais comumente se verificava o compartilhamento de códigos (geralmente na forma de código-fonte, formato legível para seres humanos) em grupos de usuários de equipamentos de um mesmo fabricante, sem restrições práticas. Nesses grupos, o código era eventualmente modificado e aperfeiçoado, e as alterações também compartilhadas. Pode-se mesmo afirmar que, naquele momento, o software era livre.

No início dos anos 1970, a própria IBM passou a vender software como um produto autônomo. Esse ato acabou por se mostrar seminal, eis que vários outros fabricantes o seguiram, de modo que, em meados da década, o software proprietário já se houera tornado comum. Isso denotou tanto uma enorme mudança de cultura entre os profissionais que trabalhavam com software quanto o início do surgimento de um grande número de empresas dedicadas ao novo negócio.

Em 1980, um jovem programador do Laboratório de Inteligência Artificial do Instituto de Tecnologia de Massachussetts (unidade conhecida em inglês como AI Lab), Richard M. Stallman, constatou que a nova impressora a laser instalada no local não funcionava com o modelo de computador com o qual ele trabalhava. Não era a primeira vez que Stallman se deparava com esse tipo de problema. Em outras

ocasiões, houvera encontrado solução: de posse do código-fonte do programa que faz a conexão entre o computador e a impressora, e fazendo uso de suas habilidades de programador, conseguiu adaptá-lo para uso em seu equipamento.

Desta feita, porém, o fabricante da impressora, tendo mudado sua política em alinhamento com os novos tempos do mercado de software, recusou-se a fornecer o código-fonte ao AI Lab. Naquele momento histórico, já estava estabelecida a percepção de que o software era tão importante economicamente quanto o hardware, e não apenas um acessório de valor quase sempre desprezível. Cláusulas de confidencialidade passaram a integrar os contratos de licenciamento de software, impedindo seu compartilhamento.

Para Stallman, no entanto, a cláusula de confidencialidade era algo completamente diferente. Era uma recusa [...] em participar de uma sociedade que, até então, havia incentivado os programadores de software a considerar programas como recursos comuns. Como um camponês cujo centenário canal de irrigação houvesse secado repentinamente, Stallman percorreu o canal em direção à sua nascente apenas para encontrar uma novíssima barragem de hidrelétrica ostentando o logotipo da Xerox. (WILLIAMS, 2010, p. 8).

Nos três anos posteriores, Stallman acompanhou a adoção cada vez mais intensa de software proprietário no AI Lab, enquanto sua recusa em assinar acordos de exclusividade ou não-compartilhamento acabou por torná-lo um excluído em seu próprio mundo. Isso, aliado ao fato de que a nova conjuntura o deixara impotente diante de problemas que outrora poderia resolver com facilidade, culminou em uma situação insustentável para ele, levando-o a deixar o AI Lab no início de 1984 e fundar o Projeto GNU. O objetivo da iniciativa de Stallman era produzir um sistema de software completo para uso geral, mas totalmente livre (GONZÁLEZ-BARAHONA; SEOANE PASCUAL; ROBLES, 2009, p. 20).

No ano seguinte (1985), devido ao envolvimento de outras pessoas no Projeto GNU<sup>3</sup> e à necessidade de financiamento de suas atividades, Stallman criou a Free Software Foundation (FSF). Parte da receita da nova fundação advinha da venda de fitas contendo um aclamado programa de edição de texto escrito pelo próprio Stallman (o Emacs) e de manuais livres (STALLMAN, 2010, p. 13).

## 2.2 Nasce um movimento

O empreendimento de Stallman deu início a uma verdadeira mobilização, a qual foi conhecida, nos seus primórdios, como movimento do software livre. Para garantir as liberdades oferecidas pelos programas componentes do Projeto GNU, Stallman criou, em 1989, a Licença Pública Geral do GNU (GNU General Public Licence, em inglês, muitas vezes referenciada pela sigla GNU GPL).

---

<sup>3</sup> Acrônimo em inglês para *GNU's not Unix* (GNU não é Unix). Unix, no caso, é o nome do *software* proprietário cujo modo de funcionamento serviu como modelo para o projeto de Stallman. Cuida-se de um acrônimo recursivo, no qual o objeto definido toma parte em sua própria definição. Conforme o relato de Williams (2010, p. 22-22), Stallman decidiu propositalmente fazer dele um termo contraditório.

Essas liberdades, no âmbito da licença GPL (FREE SOFTWARE FOUNDATION, 2012, online), são as seguintes:

- a. a liberdade de executar o programa, para qualquer propósito (liberdade nº 0);
- b. a liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades. O acesso ao código-fonte é uma condição prévia para o exercício dessa liberdade (liberdade nº 1);
- c. a liberdade de redistribuir cópias, de modo que você possa auxiliar outras pessoas (liberdade nº 2);  
a liberdade de aperfeiçoar o programa e distribuir esses aperfeiçoamentos para o público, de modo a beneficiar toda a comunidade. O acesso ao código-fonte também é uma condição prévia para o exercício dessa liberdade (liberdade nº 3).

Outros projetos de software livre, além daqueles sob o guarda-chuva da Free Software Foundation, logo adotaram a licença GNU GPL ou desenvolveram outras licenças a fim de garantir liberdades semelhantes às da GPL. No aspecto jurídico, essas licenças introduzem características próprias e peculiares ao sistema jurídico, constituindo-se também em inovação.

Dizemos que é um contrato de licenciamento em rede porque institucionaliza uma livre reprodução de inovações e de uso do software em cadeia, através do mecanismo que faz com que o licenciado de hoje seja ipso facto o licenciante de amanhã. Num certo sentido, este contrato é uma espécie de contrato viral, na medida em que a cláusula do compartilhamento obrigatório inculca-se em todos os contratos, os fazendo partícipes de uma mesma situação. (FALCÃO; SOUZA; ARGUELHES, 2007, p. 16, grifo dos autores)

Em 1998, um grupo de desenvolvedores, em resposta à liberação do código-fonte do navegador de Internet Netscape Navigator, fundou a Open Source Initiative (OSI). Essa instituição elaborou uma definição própria para o termo código-fonte aberto – open source (OPEN SOURCE INITIATIVE, 2012, online), que, por sua extensão, não cabe ser reproduzida nesta oportunidade. Em suma, os principais acréscimos em relação ao software livre são:

- a. a licença não deve impor restrições quanto ao uso do programa, admitindo, por exemplo, que o software de código aberto seja usado comercialmente e para pesquisa genética; os direitos associados ao programa não devem depender de que o programa seja parte de uma distribuição específica de programas;
- b. a licença não deve proibir que outros softwares de código não aberto sejam distribuídos em conjunto com o software de código aberto; e
- c. nenhuma cláusula da licença pode estabelecer uma tecnologia individual, estilo ou interface a ser aplicada no programa.

Para Wong e Sayo (2009, p. 7), a OSI tem por foco valores técnicos de elaboração de softwares poderosos e confiáveis, e é mais amigável ao mercado que a FSF. Ela está menos absorpta nos aspectos morais do software livre e mais nas vantagens práticas no método de desenvolvimento distribuído do software de código-fonte aberto.

Embora as definições de open source e de software livre guardem diferenças importantes entre si, ambas visam garantir a liberdade do usuário relativamente aos programas, partilham do mesmo modo de desenvolvimento e dos esforços contra o software proprietário e suas patentes. Não por outro motivo, a partir de então, o movimento tornou-se também conhecido pela sigla F/OSS (free and open source software – software livre e de código-fonte aberto).

### **2.3 Uma obra completa, mas nunca acabada**

No início da década de 1990, o Projeto GNU já havia logrado êxitos, tendo desenvolvido vários dos módulos que compõem um sistema operacional completo, seu objetivo primordial. Faltava, porém, o núcleo do sistema, denominado kernel, responsável por controlar as funções básicas do computador. Em 1991, Linus Torvalds, um jovem estudante finlandês de computação, anunciou, em um grupo de notícias, estar desenvolvendo um sistema operacional para computadores com arquitetura Intel x86 – ou seja, para os computadores pessoais mais populares à época.

O anúncio despertou interesse, de modo que, já no ano seguinte, o novo sistema operacional tornou-se utilizável graças à contribuição de outras pessoas, incentivando Torvalds a lançá-lo ao grande público nos termos da licença GNU GPL, sob o nome de Linux (MICHALEC, 2002, p. 15-16). O trabalho de Torvalds e seus colaboradores, em conjunto com as ferramentas já disponibilizadas pelo Projeto GNU, constituía enfim um sistema operacional completo e livre, que passou a ser conhecido como GNU/Linux.

O GNU/Linux continuou a ser desenvolvido e aperfeiçoado com o passar dos anos, tendo acompanhado a evolução dos processadores, dos periféricos e suportando as novas arquiteturas de computadores que vieram surgindo ao longo do tempo.

### **2.4 Um novo modelo de negócios**

Dessa forma, o movimento do software livre irrompe no seio de uma conjuntura a que se costuma denominar sociedade da informação, termo cunhado nos anos 1960 e que, a partir da década seguinte, tornou-se comum no meios científicos e não científicos para descrever a realidade hodierna, ao lado de expressões como sociedade pós-industrial e revolução dos colarinhos brancos (KARVALICS, 2007, p. 5).

Independentemente do nome que se lhe dê, o que caracteriza essa realidade é a posição central que a informação nela ocupa. No plano econômico, ela assume a função de matéria-prima, não apenas sob a forma de informação para agir sobre uma nova tecnologia (tal como nas revoluções tecnológicas antecedentes), mas, e principalmente, constituindo-se como a própria tecnologia, atuando em proveito de

si mesma (CASTELLS, 2005, p. 108). Trata-se de um fenômeno a que Söderberg (2008, p. 74-98) denominou *commodification of information*, ao descrever de que forma não apenas a informação, mas também o trabalho empenhado em produzi-la, transformam-se em mercadorias sujeitas às leis de mercado.

A inovação constante – em outras palavras, a produção contínua de novos conhecimentos a partir da informação – torna-se, muito mais que um diferencial competitivo, um fator crucial para sua própria existência das empresas. A fonte informativa primária que subsidia o processo de inovação é o consumidor, do qual se capta o desejo (muitas vezes inconsciente) ou a quem se incute uma necessidade de consumo. Nas palavras de Lazzarato e Negri (2001, p. 44), “parece então que a mercadoria pós-industrial é o resultado de um processo de criação que envolve tanto o produtor quanto o consumidor”.

Nesse cenário, a venda de mídias contendo software livre ou os respectivos manuais é atitude inovadora que desponta como estratégia de sobrevivência da Free Software Foundation e do movimento do software livre. Ao primeiro relance, vender software livre pode parecer incoerente, porque o senso comum se apegou à ideia de que “livre” é sinônimo de “grátis”. Na língua inglesa, a tendência à confusão é ainda maior, dado que a mesma expressão (*free software*) poderia designar, a um tempo, “software grátis” e “software livre”. É o próprio Stallman (2010, p. 65) quem esclarece:

A palavra “free” possui dois significados gerais válidos; pode se referir tanto a liberdade quanto a preço. Quando falamos em “free software” [software livre], estamos falando de liberdade, não de preço [...]. Especificamente, significa que um usuário é livre para executar o programa, modificá-lo e redistribuí-lo com ou sem alterações. Programas livres são distribuídos às vezes de forma gratuita, e outras vezes por um preço considerável. Não raro, o mesmo programa está disponível de ambas as formas em diferentes lugares. O programa é livre malgrado seu preço, eis que os usuários têm a liberdade de usá-lo.

Sendo assim, por quais motivos alguém estaria disposto a pagar por algo que pode obter gratuitamente? Pode-se anotar pelo menos dois deles: a comodidade – na época em que Stallman vendia as fitas com o Emacs, não era simples (embora fosse possível) obter o programa por outra forma; e o fator moral – a consciência de que o valor pago pelo software livre representa justa retribuição pelo trabalho nele contido, e ajudará a financiar seu desenvolvimento.

Para Lemos (2005, p. 71-72), o movimento do software livre é fruto da subversão das ideias tradicionais de propriedade relativas aos “bens intelectuais”, cuja raiz se encontra na relativa insatisfação quanto ao regime tradicional do direito autoral aplicado ao software, na medida em que aquele impedia as possibilidades de se atender a objetivos outros além daqueles puramente econômicos. Mais além (p. 74), esse autor afirma que, antes da existência do movimento do software livre, o desenvolvimento de softwares com base no tradicional regime de direito autoral era visto como algo inquestionável: uma instituição jurídica não poderia ser melhorada ou alterada, e a sociedade, como um todo, simplesmente aceitava seus contornos como fato estabelecido. O advento do movimento do software livre deixou

explícito que esse regime tradicional se constituía, em diversos aspectos, num obstáculo à inovação.

## 2.5 Um novo modo de produção

Com o movimento do software livre, emergia, inequivocamente, também um modo de produção, que, aliado à exploração econômica do software livre, veio instituir um novo modelo de negócios e, quiçá, também um novo paradigma econômico. Uma das mais bem sucedidas explicações desse novo modelo é a metáfora da catedral e do bazar. Nas palavras de seu autor:

Eu acreditava que os softwares mais importantes [...] necessitavam ser construídos como as catedrais, habilmente criados com cuidado por mágicos ou pequenos grupos de magos trabalhando em esplêndido isolamento [...].

O estilo de Linus Torvalds de desenvolvimento [...] veio como uma surpresa. Nenhuma catedral calma e respeitosa aqui – ao invés, a comunidade Linux pareceu assemelhar-se a um grande e barulhento bazar de diferentes agendas e aproximações [...] de onde um sistema coerente e estável poderia aparentemente emergir somente por uma sucessão de milagres. (RAYMOND, 1998, p. 2).

O modelo colaborativo acabou por se consolidar como um importante meio de produção de informação, o que é devido, segundo Benkler (2002, p. 34-37), aos atributos de uma economia da informação pervasivamente reticulada. Tal visão de economia informacional e em rede, registre-se, é também partilhada por Castells (2005, p. 119).

Os caracteres identificados por Benkler nessa economia são, pois:

- a. o objeto da produção, a informação, é um objeto peculiar na análise econômica, visto que seu consumo por alguém não diminui sua disponibilidade de uso para outrem e seu insumo primário não-humano é o mesmo bem público que seu resultado – a informação;
- b. os custos de capital material para a produção de informação caíram acentuadamente com a introdução de redes de computadores baseados em processadores mais baratos;
- c. o principal componente humano, o talento criativo, é altamente variável, mais que o trabalho tradicional, e, com certeza, mais também que muitos recursos humanos frequentemente indispensáveis à produção. Ademais, os indivíduos que fornecem esse componente humano estão mais bem informados que ninguém quanto à variabilidade e à adequação de seus talentos, níveis de motivação e foco em um dado momento, para uma dada tarefa; e
- d. a comunicação e o intercâmbio de informações através do espaço e do tempo estão mais acessíveis do que nunca, o que permite a coordenação das fontes potenciais de empenho criativo amplamente distribuídas e a agregação do esforço distribuído em produtos finais úteis.

## 2.6 Uma nova (contra)cultura

Por toda a amplitude do movimento do software livre, é possível identificar a existência de problemas relativos à adaptação externa (como as alterações de código feitas por Stallman para que as antigas impressoras funcionasse com o modelo de computador com o qual ele trabalhava) e também à integração interna (a coordenação do trabalho de diferentes colaboradores em torno de um objetivo comum). Esses problemas foram resolvidos com sucesso, quando se considera o grande número de projetos de software livre hoje existentes, sua aceitação e reputação. Logo, o padrão de pressupostos básicos foi validado, habilitando-se a ser adotado pela novas gerações. Recorde-se que o movimento do software livre nasceu da reação e do inconformismo a um estado de coisas estabelecido, caracterizando-o como um movimento de contracultura. Dessa forma, é possível traçar seus pressupostos a partir do exato oposto daqueles em que se baseia seu objeto de combate. Stallman (2010, p. 8) identifica alguns deles: (1) o pressuposto de que as empresas de software detém um direito natural e inquestionável de possuir o software e, por extensão, têm poder sobre todos os usuários; (2) o pressuposto de que tudo o que importa em relação ao software é quais tarefas ele permite a alguém cumprir, e que os usuários de computador não devem se preocupar sobre que tipo de sociedade é-lhes permitido ter; (3) o pressuposto de que não existiria software utilizável (ou jamais haveria um software capaz de fazer esta ou aquela tarefa) se não se desse a uma empresa o poder sobre os usuários do programa. Todos esses pressupostos, ressalta o autor, manifestam-se tacitamente – emergem das entrelinhas de um discurso das grandes empresas de software, quando falam sobre “fazer garantir” seus “direitos” ou, ainda, “acabar com a pirataria”.

Ter-se-ia, enfim, como pressupostos do movimento do software livre, em contraponto aos apresentados: (1') o direito de proteção à liberdade intelectual não é um direito natural, mas um monopólio imposto pelo governo que limita o direito de copiar natural dos usuários; (2') que o software, com tudo o que permite realizar (execução, estudo, alteração, compartilhamento), tem o potencial de se constituir num instrumento de atuação política, com reflexos na estrutura da sociedade; e (3') não apenas é possível, como já foi feito e demonstrado, a existência softwares úteis e livres de correntes.

O software livre ou de código-fonte aberto é, em si mesmo, a resposta a um problema. Fez frente ao desafio coletivo de criar regimes de governança que exigiam e requeriam conhecimento empírico confiável como alicerce de sua legitimação política. Essa legitimação política não é um problema eterno ou teórico, mas antes um problema de prática constante no mundo real com vistas à criação de infraestruturas pelas quais os indivíduos vêm tomar parte e entender sua própria governança, seja esta feita por estados, corporações ou máquinas. Se o poder busca o consentimento daqueles aos quais governa – e especialmente o consentimento do tipo democrático e autônomo que se tornou o ideal global desde o século XVII – deve também buscar assegurar a estabilidade e a fiabilidade do conhecimento sobre o qual esse consentimento se assenta (KELTY, 2008, p. 305).

Um aspecto saliente desse modo colaborativo de produção é que as pessoas se engajam nos

projetos e dedicam seu tempo “livre” sem esperar, em troca, remuneração ou direitos autorais. O que as move é a possibilidade de poder participar de um modelo colaborativo global e incentivos que passam ao largo de benefícios de cunho econômico, sendo mais importante para elas interesses sociais e individuais (LEMOS, 2005, p. 79-81).

Pelo visto, a motivação está na raiz do sucesso da cultura colaborativa. Para Benkler (2006, p. 133), é necessário compreender a motivação humana além do modelo simplista proposto com frequência pela ciência econômica. Esta tem por pressuposto que todas as motivações humanas podem ser reduzidas a algo como utilidades positivas ou negativas – coisas que as pessoas querem, e coisas que elas querem evitar. Isso seria passível de ser contabilizado e traduzido (quase sempre) em um meio universal de troca, o dinheiro.

No entanto, essa visão se torna inadequada e até mesmo errada em contextos não econômicos – considere-se, a título de exemplo, todo o ecossistema em torno dos bancos de sangue, no qual doadores voluntários não auferem qualquer vantagem, em termos monetários, por oferecer de forma gratuita parte de um tecido vital seu. Se o modelo colaborativo é capaz de perverter a lógica de mercado, haveria de também transformar o conceito de retribuição.

### **3 Análise sistêmica do software livre**

#### **3.1 O software livre e seu movimento enquanto sistema**

A abordagem sistêmica é um modo de apreensão e estudo da realidade, que se contrapõe ao reducionismo típico do paradigma cartesiano. Ao invés de simplificar os fenômenos, analisando-os por partes para inferir conclusões que, pretensamente, seriam aplicáveis ao todo, essa visão fia-se na concepção de sistema. Para Ackoff (apud SKYTTNER, 2005, p. 58), um sistema pode ser entendido como um conjunto de dois ou mais elementos que satisfazem três condições: (1) o comportamento de cada elemento tem efeito no comportamento do todo; (2) o comportamento dos elementos e seus efeitos no todo são interdependentes; e (3) embora se possam formar subgrupos de elementos, todos têm efeito no comportamento do todo, mas nenhum deles o faz de modo independente.

No caso do movimento do software livre, é possível reconhecer diversos elementos em interação, sendo que cada qual poderia, de forma independente, ser estudada por uma disciplina científica apartada. Há o software em si, objeto de Engenharia própria; o conjunto de princípios que informam o software livre, de fundo notadamente filosófico; o movimento iniciado por Stallman e a cultura subjacente, merecedores de uma análise sociológica; um novo modo de produção, de interesse das Ciências Econômicas, entre muitos outros. O fato é que nenhum desses elementos isoladamente, com as interpretações dadas pela respectiva especialidade científica afeta, é capaz de explicar, no todo, o movimento do software livre. Daí a necessidade do uso de uma metodologia sistêmica para compreendê-lo.

### 3.2 O Critical Systems Heuristics (CSH)

O Critical Systems Heuristics – CSH (Heurística Crítica Sistêmica) é uma metodologia sistêmica proposta por Werner Ulrich (1983) como apoio à prática reflexiva, tendo-se tornado um dos principais métodos empregados na abordagem sistêmica. Em sua forma mais simples, o CSH consiste em um conjunto de doze perguntas que explicitam os juízos cotidianos nos quais se confia (conscientemente ou não) para entender situações e projetar sistemas com vistas a melhorá-las, conforme exposto na Tabela 1.

Um dos problemas inerentes à definição dos sistemas abertos, isto é, aqueles não autocontidos, que interagem trocando elementos e informações com o ambiente do entorno (VON BERTALANFFY, 1976, p. 39), é exatamente demarcar com precisão seus limites; em outras palavras, conhecer a fronteira entre o sistema e seu ambiente, implicando, em última instância, determinar a extensão daquele. As categorias e questões propostas pelo CSH vêm em auxílio a este equacionamento, posto que respeitam “juízos limítrofes”, segundo seu autor. Tais juízos provêm de quatro diferentes fontes de influência, assim definidas e explicadas em Ulrich e Reynolds (2010, p. 259):

1. Fontes de motivação: de onde provêm o senso de intencionalidade e princípios valorativos;
2. Fontes de controle: de onde emanam os recursos e poderes necessários;
3. Fontes de conhecimento: onde se supõe que haja disponibilidade suficiente de competência e experiência; e
4. Fontes de legitimação: de onde se espera que proceda a aprovação legal e social.

**Tabela 1: Categorias limítrofes e questões do CSH**

Fontes de influência	Juízos limítrofes que informam um sistema de interesse (S)			
	<i>Papéis sociais (partes interessadas)</i>	<i>Preocupações específicas (interesses)</i>	<i>Problemas-chave (questões das partes interessadas)</i>	
<b>Fontes de motivação</b>	1. <i>Beneficiário</i> Quem deve ser/é o beneficiário desejado do sistema S?	2. <i>Objetivo</i> Qual deve ser/é o objetivo de S?	3. <i>Medida de progresso</i> Qual deve ser/é a medida de sucesso?	<b>Os envolvidos</b>
<b>Fontes de controle</b>	4. <i>Tomador de decisões</i> Quem deve estar/está no controle das condições do sucesso de S?	5. <i>Recursos</i> Quais condições de sucesso devem estar/estão sob o controle de S?	6. <i>Ambiente de decisão</i> Quais condições de sucesso devem estar/estão fora do controle do tomador de decisão?	
<b>Fontes de conhecimento</b>	7. <i>Especialista</i> Quem deve ser/é o provedor de conhecimentos relevantes e habilidades para S?	8. <i>Competência</i> Quais devem ser/são os novos conhecimentos relevantes para S?	9. <i>Fiador</i> O que deveria ser/é considerado como garantia de uma implementação bem-sucedida?	
<b>Fontes de legitimação</b>	10. <i>Testemunha</i> Quem deve estar/está representando os interesses daqueles negativamente afetados, mas não envolvidos em	11. <i>Emancipação</i> Quais devem ser/são as oportunidades para que os interesses daqueles afetados negativamente tenham expressão e	12. <i>Visão de mundo</i> Qual espaço deve estar/está disponível para a reconciliar diferentes visões de mundo acerca de S, entre os envolvidos	<b>Os afetados</b>

S?	liberdade, a partir da visão de mundo de S?	e os afetados?
----	---	----------------

Fonte: Traduzido de Ulrich e Reynolds (2010, p. 244).

Cada uma dessas fontes é perquirida por três perguntas, totalizando doze questionamentos. Por seu turno, cada questão apresenta-se sob duas formas: um modo empírico, utilizado para diagnosticar a percepção do estado atual do sistema, e no qual as perguntas utilizam o verbo no modo indicativo (“é/está”); e um modo normativo, para projetar o sistema em sua forma e funcionamento ideais, conjugando os verbos no modo potencial (“deve ser/deve estar”).

### 3.3 O método em prática: utilizando o CSH para compreender o movimento do software livre

As características do CSH o habilitam como instrumento de intervenção sobre a realidade, sendo esse seu principal propósito (ULRICH; REYNOLDS, 2010, p. 250). Sem embargo, ousamos propor seu uso para analisar acontecimentos pretéritos e presentes, interpretando intervenções já realizadas à sua luz.

Assim, passa-se à análise sistêmica do movimento do software livre, valendo-se de excertos de Stallman (2010) como corpus para responder às perguntas contextualizadas do CSH, e aponto-lhes nossos comentários (Tabela 2).

Tabela 2: Análise do movimento do software livre utilizando o método CSH

Nº	Questões do CSH		Análise
	Situação fática (“é”)	Situação desejável (“deve ser”)	
1	<p><i>Quem é beneficiado pelo software?</i></p> <p>Os autores não raro invocam uma conexão especial com os programas que escreveram, e seguem adiante afirmando que, em consequência, seus desejos e interesses em relação ao programa simplesmente são mais importantes que aqueles de outras pessoas – ou mesmo que os do resto do mundo inteiro. (Comumente, são as empresas, e não os autores, que detêm os direitos autorais sobre o <i>software</i>, mas se espera que ignoremos essa discrepância.) (p. 38)</p>	<p><i>Quem deve ser beneficiado pelo software?</i></p> <p>Uma vez o <i>software</i> GNU tenha sido escrito, todos poderão obter um bom sistema de <i>software</i> gratuitamente, exatamente como obtêm o ar. (p. 30)</p>	<p>Stallman pondera que os beneficiários do <i>software</i>, enquanto proprietário, são seus autores (pessoas físicas), ou, antes e mais comumente, a indústria de <i>software</i>, que costuma deter os direitos autorais dos programas produzidos às suas expensas. Defende ele que os beneficiários deveriam ser, ao invés, os usuários do <i>software</i>.</p>
2	<p><i>Qual é o objetivo do software?</i></p> <p>O sistema de direitos autorais confere “donos” aos programas de <i>software</i>, muitos dos quais visam renegar os benefícios potenciais do <i>software</i> ao resto das pessoas. Eles gostariam de ser os únicos capazes de copiar e modificar o <i>software</i> que usamos. (p. 37)</p>	<p><i>Qual deve ser o objetivo do software?</i></p> <p>O objetivo do GNU foi dar liberdade aos usuários, não apenas ser popular. Dessa forma, precisamos usar termos de distribuição que evitassem que <i>software</i> GNU fosse transformado em <i>software</i> proprietário. (p. 12)</p>	<p>O objetivo do <i>software</i> proprietário é garantir o lucro dos detentores do respectivo direito autoral, mediante o cumprimento deste. Isso implica direcionar os benefícios apenas a alguns, em detrimento da maioria. Stallman, por intermédio da licença GNU, deseja garantir a liberdade dos usuários sobre o <i>software</i>, democratizando suas benesses.</p>
3	<p><i>Qual é a medida de sucesso de um software?</i></p> <p>O exemplo paradigmático [...] é o X Window System. Desenvolvido no MIT, e lançado como <i>software</i> livre com</p>	<p><i>Qual deve ser a medida de sucesso de um software?</i></p> <p>Contudo, o movimento do <i>software</i> livre é um movimento social, não um mercado, e o sucesso a que almeja não</p>	<p>O sucesso do <i>software</i> proprietário é medido por sua adoção em larga escala, o que geralmente se traduz em alta lucratividade. Stallman defende que o <i>software</i> livre e o movimento dele originado serão bem sucedidos na</p>

<p>uma licença permissiva, foi logo adotado por várias empresas de computadores. Elas adicionaram o X aos seus sistemas proprietários Unix, somente na forma binária<sup>4</sup>, e coberto por uma cláusula de confidencialidade [...]. Os desenvolvedores do X Window System não reputaram isso como problema – eles esperavam e torciam para que isso acontecesse. Seu objetivo não era a liberdade, mas apenas “sucesso”, definido como “ter muitos usuários”. (p. 12)</p>	<p>é o sucesso comercial. Tentamos servir ao público dando-lhe liberdade – não competindo para tomar o mercado de um concorrente. Igualar essa campanha por liberdade a um mero esforço comercial por sucesso é negar a importância da liberdade e legitimar o <i>software</i> proprietário. (p. 98)</p>	<p>medida em que conseguirem dar liberdade ao usuário.</p>
<p>4 <i>Quem está no controle das condições do sucesso do software?</i></p> <p>Liberdade é ser capaz de tomar decisões que afetam principalmente a você; poder é ser capaz de tomar decisões que afetam aos outros mais que a você mesmo. Se confundirmos poder com liberdade, fracassaremos em defender a liberdade real. Fazer de um programa proprietário é um exercício de poder. (p. 247)</p> <p>Legisladores influenciados pelo termo “propriedade intelectual” [...] a transformaram num esquema que incentiva a publicidade comercial. (p. 90)</p>	<p><i>Quem deve estar no controle das condições do sucesso do software?</i></p> <p><i>Software</i> livre, <i>software</i> que respeita a liberdade, significa que todo usuário do programa é livre para obter o código-fonte do programa e alterá-lo de forma a que faça o que o usuário queira, e também que é livre para oferecer gratuitamente ou vender cópias, na forma original ou modificada. Isso quer dizer que os usuários estão no controle. Com os usuários no controle do <i>software</i>, ninguém tem o poder de impor funções desagradáveis a outros. (p. 240)</p>	<p>As condições do sucesso do <i>software</i> proprietário estão nas mãos da indústria de <i>software</i>. Esta manipula exercendo o poder sobre os usuários, não raro com o uso de publicidade massiva. Para as condições de sucesso do <i>software</i> livre, em contraste, são afetadas aos usuários, tanto quanto eles tenham liberdade de manipular o código-fonte, alterá-lo e depois distribuí-lo gratuitamente ou mesmo vendê-lo.</p>
<p>5 <i>Quais condições de sucesso estão sob o controle dos usuários de software?</i></p> <p>[...] o direito autoral não é um direito natural, mas um monopólio artificial imposto pelo governo, e que limita o direito natural de cópia por parte dos usuários. (p. 8)</p>	<p><i>Quais condições de sucesso devem estar sob o controle dos usuários de software?</i></p> <p>A liberdade de estudar como o programa funciona, e modificá-lo para que faça o que você queira (liberdade 1). O acesso ao código-fonte é pré-condição para isso. [...] A liberdade de distribuir cópias de suas versões modificadas a outros (liberdade 3). Ao fazer isso, você dará uma oportunidade a que toda a comunidade se beneficie de suas alterações. O acesso ao código-fonte é pré-condição para isso. (p. 3)</p>	<p>A condição de sucesso do <i>software</i> proprietário é a eficiência da aplicação do direito autoral subjacente, assegurando que a ninguém é lícito obter uma cópia do programa sem pagar, em contrapartida, o valor por ele fixado. O <i>software</i> livre depende, para seu sucesso, apenas da garantia de que seus usuários tenham acesso ao código-fonte.</p>
<p>6 <i>Quais condições de sucesso estão fora do controle dos usuários de software?</i></p> <p>Produtores, com frequência, se referem ao ato de copiar o que eles não aprovam como “pirataria”. Dessa forma, insinuam que isso é eticamente equivalente a atacar navios em alto mar, sequestrar e assassinar as pessoas a bordo. Com base nessa propaganda, eles têm conseguido aprovar leis na maior parte do mundo para proibir a cópia na maioria das (ou, às vezes, todas as) circunstâncias. (Eles ainda pressionam para tornar</p>	<p><i>Quais condições de sucesso devem estar fora do controle dos usuários de software?</i></p> <p>Com os usuários no controle do <i>software</i>, ninguém tem o poder de impor funções desagradáveis a outros. Mesmo que você não exerça esse controle pessoalmente, você faz parte de uma sociedade na qual outros o fazem. Se você não é um programador, outros usuários do programa o são. (p. 240)</p>	<p>Para a indústria do <i>software</i> proprietário, a total coibição da pirataria é algo que, apesar de seus esforços, está ainda longe de ser conseguido. No caso do <i>software</i> livre, em que os tomadores de decisão são os usuários finais, Stallman afirma que muitos deles não teriam conhecimento técnico para impedir que uma funcionalidade não desejável fosse implementada no programa. Isso é mitigado, segundo ele, pelo fato de que alguns usuários do programa são programadores, detentores das habilidades necessárias para intervir</p>

<sup>4</sup> Isto é, em formato legível apenas por máquinas, não por seres humanos.

	essas proibições mais abrangentes.) (p. 99)		no funcionamento do <i>software</i> .
7	<p><i>Quem é o provedor de conhecimentos relevantes e habilidades para o software?</i></p> <p>Outra suposição não declarada é a de que a única coisa que importa em relação ao <i>software</i> é quais tarefas ele lhe permite fazer – que nós, usuários de computador, não devemos nos preocupar com o tipo de sociedade nos é permitido ter. (p. 8)</p> <p>O <i>software</i> proprietário rejeita sua sede por conhecimento; ele diz: “O conhecimento que você quer é secreto – é proibido aprender!” (p. 58)</p>	<p><i>Quem deve ser o provedor de conhecimentos relevantes e habilidades para o software?</i></p> <p>O <i>software</i> livre contribui para o conhecimento humano, ao contrário do <i>software</i> não-livre. As universidades deveriam, portanto, incentivar o <i>software</i> livre em nome do avanço do conhecimento humano, tanto quanto devem encorajar os cientistas e outros pesquisadores a publicar seus trabalhos. (p. 59)</p>	<p>Para o sistema de <i>software</i> proprietário, na ótica de Stallman, pouco importa o conhecimento que seus usuários terão sobre o <i>software</i>, desde que eles consigam executar as tarefas para os quais este foi projetado – e nada mais. O <i>software</i> livre, por seu turno, representaria um efetivo contributo para o desenvolvimento do conhecimento humano.</p>
8	<p><i>Quais são os novos conhecimentos relevantes em relação ao software?</i></p> <p>Em qualquer campo do saber, pode-se alcançar maiores altitudes apoiando-se sobre os ombros de outros. Todavia, isso não é mais possível, de forma geral, na área de <i>software</i> – você somente pode se apoiar nos ombros de outras pessoas <i>dentro de sua própria empresa</i>. (p. 50, grifo do autor)</p>	<p><i>Quais devem se os novos conhecimentos relevantes para em relação ao software?</i></p> <p>O <i>software</i> livre incentiva todos a aprender. A comunidade do <i>software</i> livre rejeita o “sacerdócio do conhecimento”, que mantém o grande público na ignorância de como a tecnologia funciona; nós encorajamos os estudantes de qualquer idade e condição a estudar o código-fonte e aprender tanto quanto queiram saber. (p. 58)</p>	<p>Na indústria do <i>software</i> proprietário, o conhecimento sobre a estrutura e o funcionamento internos do programa são considerados segredos comerciais; novos conhecimentos baseados nesses segredos são restritos à ao ambiente da empresa, e mantêm o caráter confidencial daqueles que lhe deram origem. Por outro lado, o movimento do <i>software</i> livre estimula o estudo e a criação de novos conhecimentos baseados nos programas livres e sua respectiva divulgação ao grande público.</p>
9	<p><i>O que é considerado como garantia de uma implementação bem-sucedida?</i></p> <p>A primeira suposição é a de que as empresas de <i>software</i> tem um direito natural e inquestionável de possuir o <i>software</i> e assim ter poder sobre todos os respectivos usuários [...]. Outra suposição [...] é a de que a única coisa que importa em relação ao <i>software</i> é quais tarefas ele lhe permite fazer – que nós, usuários de computador, não devemos nos preocupar com o tipo de sociedade nos é permitido ter. Uma terceira suposição é a de que não teríamos qualquer <i>software</i> utilizável [...] se não déssemos a uma empresa o poder sobre os usuários do programa. (p. 8)</p>	<p><i>O que deveria ser considerado como garantia de uma implementação bem-sucedida?</i></p> <p>[...] Todo programa é constantemente ameaçado por patentes de <i>software</i> [...]. Para evitar isso, a GPL garante que patentes não possam ser utilizadas para tornar o programa não-livre. (p. 172)</p>	<p>A garantia de sucesso do <i>software</i> proprietário reside em sua capacidade de doutrinar seus usuários para que lhe são impostos, tudo o que é sublimado na figura das patentes. O movimento do <i>software</i> livre, por intermédio da licença GPL, visa garantir a liberdade dos usuários impedindo que o <i>software</i> licenciado nos termos daquele seja tornado proprietário por força de patentes.</p>
10	<p><i>Quem está representando os interesses daqueles negativamente afetados, mas não envolvidos com o software?</i></p> <p>Às vezes, regulações governamentais de controle de exportação e sanções comerciais podem restringir sua liberdade de distribuir cópias de programas intencionalmente. Desenvolvedores de <i>software</i> não têm o poder de eliminar ou passar sobre</p>	<p><i>Quem deve estar representando os interesses daqueles negativamente afetados, mas não envolvidos com o software?</i></p> <p>[...] o que eles [desenvolvedores de <i>software</i>] podem e devem fazer é se recusar a impor essas restrições como condição de uso do programa. Dessa maneira, as restrições não irão afetar atividades e pessoas fora da jurisdição desses governos. Portanto, licenças de</p>	<p>Tanto no caso do <i>software</i> proprietário quanto no caso do <i>software</i> livre, pessoas situadas fora da jurisdição para a qual o modelo de licenciamento foi concebido podem ver seus direitos restringidos. No que tange ao <i>software</i> proprietário, um usuário poderia se ver na estranha situação de ter pago para utilizar um programa mas não poder fazê-lo legalmente. Não haveria ninguém a defender esse usuário – nem mesmo o</p>

<p>essas restrições [...]. (p. 5)</p>	<p><i>software</i> livre não devem exigir obediência a nenhuma regulação de exportação como condição de qualquer das liberdades fundamentais. (p. 5)</p>	<p>fabricante do <i>software</i>, que não pode passar por sobre a legislação de onde atua. O <i>software</i> livre, para garantir que seu uso seja irrestrito independentemente da jurisdição, deve adotar licenças desvinculadas de normas de governos locais, em especial daquelas relativas à regulação de exportação. O papel de porta-voz dos potenciais afetados negativamente cabe à licença do <i>software</i> e, de forma mediata, à comunidade que a suporta.</p>
<p>11 <i>Quais são as oportunidades para que os interesses daqueles afetados negativamente tenham expressão e liberdade, a partir da visão de mundo subjacente ao software?</i></p> <p>A hostilidade pública em relação ao DRM<sup>5</sup> está crescendo lentamente, obstada desde que expressões de propaganda como “proteger os autores” e “propriedade intelectual” convenceram os leitores de que seus direitos não vêm ao caso. Tais termos assumem implicitamente que os editores merecem um poder especial em nome dos autores, que nós somos moralmente obrigados a nos curvar diante deles, e que prejudicamos alguém se vemos ou ouvimos algo sem ter pago pela respectiva permissão. (p. 124)</p> <p>É desanimador morar em uma casa a qual você não pode modificar para atender às suas necessidades. (p. 49).</p>	<p><i>Quais devem ser as oportunidades para que os interesses daqueles afetados negativamente tenham expressão e liberdade, a partir da visão de mundo subjacente ao software?</i></p> <p>A liberdade de estudar como o programa funciona, e modificá-lo para que faça o que você queira (liberdade 1). O acesso ao código-fonte é pré-condição para isso. (p. 3)</p>	<p>O sistema em torno do <i>software</i> proprietário oferece pouquíssimas (não raro, quaisquer) oportunidades para que os usuários insatisfeitos se manifestem. Além disso, por meio de propaganda, moldam a opinião pública de modo a que esta acredite que não tem direito algum a expressar sua insatisfação. O <i>software</i> livre dá aos usuários a chance de modificar o programa caso este não atenda às suas necessidades.</p>
<p>12 <i>Qual espaço está disponível para a reconciliar diferentes visões de mundo acerca do software, entre os envolvidos e os afetados?</i></p> <p>A competição torna-se embate quando os competidores começam a tentar atrapalhar uns aos outros, ao invés de ir em frente – quando “que vença o melhor” dá lugar a “que eu vença, sendo ou não o melhor”. O <i>software</i> proprietário é danoso, não por ser uma forma de competição, mas porque se constitui numa forma de embate entre os cidadãos de nossa sociedade. (p. 54)</p>	<p><i>Qual espaço deve estar disponível para a reconciliar diferentes visões de mundo acerca do software, entre os envolvidos e os afetados?</i></p> <p>Nós [do movimento do <i>software</i> livre] discordamos do pessoal do <i>open source</i> em valores e objetivos básicos, mas tanto a visão deles quanto a nossa leva, em muitos casos, ao mesmo comportamento prático – tal como desenvolver <i>software</i> livre. Como resultado, pessoas do movimento do <i>software</i> livre e do <i>open source</i> frequentemente trabalham juntos em projetos práticos como o desenvolvimento de <i>software</i>. É notável que essas diferentes perspectivas filosóficas possam com tanta frequência motivar diferentes pessoas a participar dos mesmos projetos. (p. 86)</p>	<p>No mundo do <i>software</i> proprietário, quase nenhuma conciliação de visões de mundo é possível, eis que está em jogo o lucro, pelo qual se combate. O lado do <i>software</i> livre, embora não isento de diferenças filosóficas muitas vezes mesquinhas, abre um espaço conciliatório quando o objetivo entre grupos divergentes é comum.</p>

<sup>5</sup> Acrônimo em inglês para *Digital Rights Management* (Gerenciamento de Direitos Digitais): “consiste em restringir a difusão por cópia de conteúdos digitais ao mesmo tempo em que se assegura e administra os direitos autorais e suas marcas registradas, pelo ângulo do proprietário dos direitos autorais” (WIKIPÉDIA, 2012, online).

Fonte: Ulrich e Reynolds (2010) – perguntas; Stallman (2010) – respostas; análise pelos autores do artigo.

#### 4 Considerações finais

O software livre, ou de código-fonte aberto, representa uma nova forma de ver os elementos de uma sociedade predominantemente capitalista. A partir de institutos jurídicos consagrados, como o direito de propriedade intelectual, e por meio da inovação, aponta as possibilidades de se trabalhar e de se produzir bens de cunho informacional de outra maneira, demonstrando que a lógica do mercado não é absoluta nem a única plausível. O potencial transformador da filosofia que suporta o movimento do software livre, aplicado também a outros produtos de informação além do software, semeia e dissemina uma cultura sem óbices, cujo fim último é retomar o controle das grandes empresas e devolvê-lo às pessoas.

A partir da análise sistêmica realizada com o uso do CSH, fica evidente que o sistema identificado como “software livre” não se encontra ainda em sua forma ideal, vislumbrada por Stallman, seu idealizador. Isso parece, porém, ser um problema mais afeto à escala que à viabilidade, eis que alguns subsistemas que o integram foram mais bem sucedidos que outros em se aproximar da forma de funcionamento proposta. As ideias de Stallman, conhecido por suas posições radicais, não foram capazes de eliminar a indústria do software proprietário nem de se estabelecer como consenso na comunidade: dissidências surgiram, como a Open Source Initiative. No entanto, o pensamento stallmaniano vêm influenciando esse mercado (e também outros), tendo já conseguido um bom número de êxitos, e as divergências internas são postas de lado em nome de um objetivo maior. Sem dúvidas, o software livre veio para mudar as regras do jogo.

Em suma, podem-se frisar os pontos seguintes:

1. o software deve existir em benefício daqueles que o utilizam;
2. seu objetivo não há de ser outro senão garantir a liberdade dos usuários;
3. será ele bem-sucedido tanto quanto conseguir garantir essa liberdade;
4. o sucesso do software não pode depender de ninguém mais além dos próprios usuários;
5. para promover o bom êxito do software, é imprescindível que os usuários tenham acesso ao respectivo código-fonte;
6. a assertiva anterior continua válida, mesmo ponderando que parcela dos usuários não é constituída por programadores (e, portanto, não saberiam o que fazer com o código-fonte), porque estes podem atuar em favor daqueles;
7. o software, na forma de código-fonte, é um rico depósito de conhecimento humano, o qual deve ser partilhado para o benefício de todos;
8. o software, por suas características já descritas, catalisa a produção de novos conhecimentos fundados em seu estudo, utilização e modificação;

9. o software será reputado como livre na medida de sua capacidade de sê-lo e de se manter nesse estado;
10. os programadores de software livre devem ter o cuidado de não liberá-lo sob licenças cujos termos venham a impedir seu livre uso, estudo, modificação e redistribuição;
11. a liberdade de modificação do software, garantida pelo acesso ao código-fonte, configura também uma oportunidade para sanar as insatisfações relativas a ele; e
12. eventuais divergências de cunho filosófico ou terminológico, no seio do movimento, são passíveis de superação quando o objetivo a ser alcançado é maior e comum.

Registre-se, enfim, que a mensagem abstraída de todo o movimento do software livre encontra eco numa uma velha máxima, com roupagem, porém, renovada: “um outro mundo [este, o sistema por excelência] é possível”.

### **Referências bibliográficas**

BENKLER, Yochai. Coase's Penguin, or, Linux and The Nature of the Firm. *Yale Law Journal*, n. 112, Aug. 2002, p. 1-73. Disponível em: <<http://www.benkler.org/CoasesPenguin.PDF>>. Acesso em: 20 jul. 2012.

\_\_\_\_\_. The wealth of networks: how social production transforms markets and freedom. 2006. Disponível em: <[http://www.benkler.org/Benkler\\_Wealth\\_Of\\_Networks.pdf](http://www.benkler.org/Benkler_Wealth_Of_Networks.pdf)>. Acesso em: 16 jul. 2012.

CASTELLS, Manuel. A sociedade em rede. Trad. por Roneide Venancio Majer. 8. ed. São Paulo: Paz e Terra, 2005, v. 1.

FALCÃO, Joaquim; SOUZA, Carlos Affonso Pereira de; ARGUELHES, Diego Werneck. A Constituição e o Software Livre. In: \_\_\_\_\_; LEMOS, Ronaldo; FERRAZ JUNIOR, Tércio Sampaio (Coords.). *Direito do software livre e a administração pública*. Rio de Janeiro : Lumen Juris, 2007, p. 9-82.

FREE SOFTWARE FOUNDATION. O que é software livre? 2012. Disponível em: <<http://www.gnu.org/philosophy/free-sw.html>>. Acesso em: 21 jul. 2012.

GONZÁLEZ-BARAHONA, Jesús M.; SEOANE PASCUAL, Joaquín; ROBLES, Gregorio. Free Software. In: HERNÁNDEZ, Jordí Mas; JIMÉNEZ, David Megías (Coords.). *Introduction to Free Software*. 3. ed. Barcelona : Fundació per a la Universitat Oberta de Catalunya, 2009, p. 9-202. Disponível em: <[http://ftacademy.org/files/materials/fta-m1-intro\\_to\\_FS-v1.pdf](http://ftacademy.org/files/materials/fta-m1-intro_to_FS-v1.pdf)>. Acesso em: 18 jul. 2012.

GIL, Antonio Carlos. Métodos e técnicas de pesquisa social. São Paulo : Atlas, 1999.

KARVALICS, László Z. Information Society – what is it exactly? (The meaning, history and conceptual framework of an expression). Budapest : Network for Teaching Information Society, 2007. Disponível em: <[http://www.ittk.hu/netis/doc/ISCB\\_eng/02\\_ZKL\\_final.pdf](http://www.ittk.hu/netis/doc/ISCB_eng/02_ZKL_final.pdf)>. Acesso em: 20 fev. 2012.

KELTY, Christopher M. Two bits: the cultural significance of free software. Durham, NC : Duke University Press, 2008. Disponível em: <<http://twobits.net/pub/Kelty-TwoBits.pdf>>. Acesso em: 2 jul. 2012.

LAZZARATO, Maurizio; NEGRI, Antonio. Trabalho imaterial: formas de vida e produção de subjetividade. Trad. de Mônica Jesus. Rio de Janeiro : DP&A, 2001.

LEMOS, Ronaldo. Direito, tecnologia e cultura. Rio de Janeiro : Editora FGV, 2005.

MICHALEC, Greg. Free software: history, perspectives, and implications. 2002, 110 p. Thesis (Bachelor Philosophy Interdisciplinary Studies) – Miami University, Oxford, OH, 2002. Disponível em: <<http://greg.primate.net/sp/thesis.pdf>>. Acesso em: 17 jul. 2012.

OPEN SOURCE INITIATIVE. The Open Source Definition. 2012. Disponível em: <<http://opensource.org/docs/osd>>. Acesso em: 21 jul. 2012.

RAYMOND, Eric S. A catedral e o bazar. Traduzido por Erik Kohler. [S. l. : s. n], 1998. Disponível em: <<http://rede.acesasp.sp.gov.br/sites/default/files/catedral-bazar.pdf>>. Acesso em: 19 jul. 2012.

SKYTTNER, Lars. General systems theory: problems, perspectives, practice. 2. ed. Singapore : World Scientific, 2005.

SÖDERBERG, Johan. Hacking capitalism: the free and open source software movement. New York, NY : Routledge, 2008.

STALLMAN, Richard M. Free Software, Free Society: selected essays of Richard M. Stallman. 2. ed. Boston, MA : The Free Software Foundation, 2010. Disponível em: <<http://www.gnu.org/doc/fsfs-ii-2.pdf>>. Acesso em: 18 jul. 2012.

ULRICH, Werner. Critical Heuristics of Social Planning: A New Approach to Practical Philosophy.

Bern/Stuttgart: Haupt. 1983.

\_\_\_\_\_; REYNOLDS, Martin. Critical system heuristics. In: REYNOLDS, Martin; HOLWELL, Sue (Eds.). *Systems Approaches to Managing Change: A Practical Guide*. London: Springer, 2010, p. 243–292. Disponível em: <[http://oro.open.ac.uk/21299/1/systems-approaches\\_ch6.pdf](http://oro.open.ac.uk/21299/1/systems-approaches_ch6.pdf)>. Acesso em: 17 ago. 2012.

VON BERTALANFFY, Ludwig. *Teoría general de los sistemas: fundamentos, desarrollo, aplicaciones*. Traducción de Juan Almela. Ciudad de México : Fondo de Cultura Económica, 1976.

WIKIPÉDIA. *Gestão de direitos digitais*. Disponível em: <[http://pt.wikipedia.org/wiki/Gest%C3%A3o\\_de\\_direitos\\_digitais](http://pt.wikipedia.org/wiki/Gest%C3%A3o_de_direitos_digitais)>. Acesso em: 31 ago. 2012.

WILLIAMS, Sam. *Free as in Freedom (2.0): Richard Stallman and the Free Software Revolution*. Boston, MA : The Free Software Foundation, 2010. Disponível em: <<http://static.fsf.org/nosvn/faif-2.0.pdf>>. Acesso em: 18 jul. 2012.

WONG, Kenneth; SAYO, Phet. *Free/Open Source Software: a general introduction*. Kuala Lumpur: The United Nations Development Programme's – Asia-Pacific Development Information Programme, 2004. Disponível em: <[http://www.iosn.net/foss/foss-general-primer/foss\\_primer\\_print\\_covers.pdf](http://www.iosn.net/foss/foss-general-primer/foss_primer_print_covers.pdf)>. Acesso em: 17 jul. 2012.