

Software Livre e Sustentabilidade *Open Source Software and Sustainability*

SCHIAVONI, Flávio Luiz

Doutor em Ciências, Universidade Federal de São João del Rei, fls@ufsj.edu.br

RESUMO

Este artigo traz uma proposta de incluir a tecnologia como um dos itens necessários para a existência humana nos dias de hoje e, por isto, um item que requer uma análise pelo ponto de vista da sustentabilidade. Mais precisamente, a tecnologia acessível por meio dos computadores e dos sistemas e aplicações neles existentes. Diante de tal necessidade, o artigo defende que o software livre é a opção mais sustentável para a adoção tecnológica. Neste ponto, o artigo apresenta o conceito de software livre, como funciona seu desenvolvimento e a diferença entre software livre e software gratuito e confronta estes conceitos com os conceitos de sustentabilidade.

PALAVRAS-CHAVE: Software livre, Open Source Software, Sustentabilidade.

ABSTRACT

This paper brings a purpose to include the technology as a necessary item to human existence in our time and, for this reason, an item that requires a analysis through the sustainability point of view. More precisely, the technology accessible by computers, computer systems and applications. Faced with such need, the paper defends the open source software as the more sustainable option for a technological adoption. At this point, the paper presents the open source software concepts, how it is the development of this kind of software and the difference between open source software and free software putting side by side these software concepts and sustainability.

KEY-WORDS : *Open source software, free software, sustainability.*

1 INTRODUÇÃO

Ao citar os itens importantes para a vida no planeta, Boff [2012, pg 10] cita a água, ar, solo, biodiversidade, florestas e energia mas termina o prefácio de seu livro apontando que devemos nos preocupar com a sustentabilidade em todos os campos da realidade. Um dos campos desta realidade é o tecnológico, campo que envolve a tecnologia, computação e software e que raramente somos convidados a pensar sob a ótica da sustentabilidade.

A computação, como uma área transversa e ortogonal ao conhecimento humano vem se tornando essencial para todas as áreas do saber. A capacidade de armazenamento tornou o computador a memória presente, suas redes tornou-o imprescindível para a comunicação humana, sua interface e capacidade de processamento tem tornado-o cada vez mais onipresente ou ubíquo, sua capacidade de digitalização de imagens e sons o tornou a mídia pela qual passa a cultura e a arte. Mesmo assim, ainda não consideramos o acesso a tecnologia computacional como um item de necessidade básico para a sobrevivência humana e pouco questionamos o ecossistema computacional que utilizamos diariamente sob a égide da sustentabilidade.

Herbert Marshall McLuhan escreveu que “O meio é a mensagem”. Se a mensagem passa pelo software, este é o meio. E se o meio é a mensagem, da liberdade do software dependerá a liberdade da mensagem. Usamos meios computacionais para nos comunicarmos diariamente mas poucas vezes nos preocupamos com a sustentabilidade deste sistema e a interação entre nosso universo e o sistema computacional que é hoje o meio de nossas mensagens.

Já BROCHI [2008, pg 28] comenta que não apenas o conteúdo é decisivo para a sustentabilidade mas também sua forma quando pensamos na cultura de sustentabilidade como um sistema auto valorado humano. Ele apresenta uma indagação se a forma é aberta ou fechada, independentemente do conteúdo. Porém, ao usarmos sistemas computacionais em nossas vidas raramente perguntamos sobre a forma destes sistemas ou se os mesmos são abertos ou fechados.

Já Flusser [1982] abre um questionamento quanto às ferramentas humanas:

Com a revolução industrial, esta situação se transforma. A informação não mais é diretamente impressa sobre pedaços do mundo, mas passa pelo crivo da ferramenta. O sapateiro não mais imprime a sua ideia do sapato sobre o couro, mas o engenheiro imprime tal ideia sobre a ferramenta, que a imprime sobre o couro. A ferramenta contém doravante o modelo do sapato, da casa pré-fabricada.

Se a ferramenta imprime sua mensagem no objeto que ela constrói, para termos real controle sobre a construção do objeto precisamos entender a construção da ferramenta. O quanto de acesso temos à construção de nossas ferramentas pode nos ajudar a entender sua construção.

Neste artigo, tais questionamentos são apresentados juntamente com a propostas de que a adoção do Software Livre é a opção mais viável atualmente para aliar desenvolvimento tecnológico com sustentabilidade.

2 SOFTWARE LIVRE

A computação é muito nova e por isto muitos dos personagens de sua história estão presentes para apresentarem suas versões dos fatos. No documentário *“Triumph of the Nerds: The Rise of Accidental Empires”* [SEN, GAU, SEGALLER, CRINGELY, CRINGELY, 1996], diversos personagens que fazem parte da história da computação comentam que no início da história da computação, principalmente a computação doméstica, boa parte dos envolvidos em projetos de software amador trabalhavam e trocavam suas criações e experiências sem preocupações com direito autoral ou patentes.

Tal realidade passou por uma mudança quando em 1976 Bill Gates escreve uma carta intitulada *“Open Letter to Hobbyists”* [GATES, 1976] reclamando infringimento de direitos autorais por parte da comunidade hobbista de software principalmente em relação aos produtos desenvolvidos por sua empresa, a Microsoft. Na carta, ele diz que a cópia não autorizada de sua linguagem de programação, o BASIC, poderia desencorajar os desenvolvedores a investir seu tempo e dinheiro criando software de qualidade sem receber por isto. Esta carta é considerada por muitos o início do mercado de software no mundo.

Se neste momento Bill Gates cria o movimento do software pago e cujo o código-fonte é restrito ao seu desenvolvedor, anos depois Richard Stallman [1999] inicia um movimento contrário que pede liberdade ao software e livre acesso ao seu código. Stallman iniciou em 1983 um movimento para que os usuários de

computadores pudessem ter a liberdade de compartilhar e melhorar o software que eles utilizam.

O conceito de software livre criado por Stallman baseia-se em 4 liberdades:

0. A liberdade de executar o programa, para qualquer propósito.
1. A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. Acesso ao código-fonte é um pré-requisito para esta liberdade.
2. A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.
3. A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para esta liberdade.

É importante notar que as liberdades propostas por Stallman e presentes no chamado Software Livre diz respeito ao uso do software e de seu código e não diz respeito a sua distribuição gratuita. Esta é talvez um dos grandes equívocos em relação ao software livre (*Free Software* em inglês). A palavra “free” em inglês significa tanto livre quanto gratuito, permitindo equivocadamente chamar de “livre” um software que é gratuito porém fechado - cujo fabricante não permite acessar seu código-fonte. Certamente, o oposto de software livre não é software pago mas software fechado.

2.1 O que significa o acesso ao código-fonte?

A liberdade de acessar o código-fonte de um programa traz algumas implicações que não são óbvias à primeira vista. Praticamente todo software é escrito em uma linguagem de programação, uma linguagem próxima da linguagem humana, chamado código-fonte. Este código é passível de leitura e entendimento por programadores. Para ser executado por computadores ele passa por um processo chamado compilação, que transforma este código em uma linguagem de máquina, também chamado de código executável. Este código compilado não é simples de ser lido e costuma ser entendido apenas por máquinas ou usuários muito avançados.

O acesso ao código fonte traz a possibilidade de aprender com este código, aperfeiçoá-lo e criar novas aplicações baseados no mesmo. Podemos com isto

reaproveitar este conhecimento e iniciar um processo de criação baseado no experimentalismo ou mesmo um processo de criação baseado na transformação.

Além disto, o acesso a este código traz também a possibilidade de auditar uma aplicação. A leitura do código antes de gerar o código de máquina pode garantir que uma aplicação não possui falhas de segurança, sejam elas acidentais ou intencionais. Por esta razão, um software livre é auditável e considerado mais seguro por não possuir falhas simples de ser encontradas.

Como o código é acessível, é inútil inserir na aplicação travas contra cópias ou verificação de compra pois um programador poderia ler o código, removê-lo e gerar uma aplicação que permite sua cópia. Por esta razão, software livre costuma também ser gratuito não sendo esta uma característica obrigatória.

2.2 Software Livre e Software Gratuito

Há milhares de projetos de software livre disponíveis na Internet. O mais famoso deles é talvez o Sistema operacional GNU/Linux. Além deste sistema operacional, há também editores de documentos como o LibreOffice, navegadores como o FireFox, Linguagens de programação, Servidores de mídia, ferramentas para design e outras. É possível que sempre exista uma ferramenta livre similar a toda ferramenta computacional paga ou de código fechado.

Há também uma gama enorme de aplicações que são distribuídas de maneira gratuita. Se estas ferramentas não distribuem seu código-fonte, elas não podem ser chamadas de software livre. Isto inclui serviços de Internet gratuitos, como Google, Facebook ou outros portais de conteúdo que não cobram para que as pessoas tenham acesso aos mesmos. Estas ferramentas podem possuir algoritmos ocultos que utilizam os dados de seus cliente para benefício próprio, nem o consentimento do usuário.

2.3 Desenvolvimento de software livre

Normalmente um software livre é mantido por uma comunidade. O conceito de comunidade de software livre é bastante interessante e permite diversos arranjos e combinações. Normalmente, uma comunidade significa um ambiente onde as pessoas conseguem interagir diretamente como uma lista de discussão de e-mail ou sala de discussão em um canal de IRC .

Há software livre que possui em sua comunidade um grande número de usuários e um grupo pequeno de desenvolvedores ou até mesmo um único desenvolvedor. Neste caso, a comunidade interage com o desenvolvedor para ajudá-lo a guiar o desenvolvimento sem contribuir efetivamente com o código, como é o caso do ambiente de programação musical Pure Data.

Há comunidades mantidas por empresas, como é o caso da distribuição de Linux Ubuntu ou Red Hat, onde a empresa mantém funcionários para o desenvolvimento da aplicação mas estes funcionários interagem com usuários ou desenvolvedores de outras empresas na comunidade.

Há também projetos que são encabeçados por universidades e centros de pesquisa, como a distribuição de aplicativos Linux Planet CCRMA mantida pelo *Center for Computer Research in Music and Acoustics* da Universidade de Stanford, por onde passam diversos alunos / pesquisadores de diversas áreas. Tais pesquisadores podem, por meio da comunidade, integrar sua pesquisa com a pesquisa de outras Universidades e permitir a colaboração interinstitucional no desenvolvimento de tecnologia garantindo que todos têm acesso ao código-fonte e que todas os pesquisadores possuem acesso total ao desenvolvimento da ferramenta.

Há comunidades com apenas desenvolvedores, comunidades pequenas até comunidades enormes que envolvem grandes empresas como é a comunidade de desenvolvimento do Linux, que inclui as empresas Google, Red Hat, IBM, Intel, AMD, entre outras [KROAH-HARTMAN, CORBET, MCPHERSON, 2009] [GONZALEZ-BARAHONA, 2013].

3 SOFTWARE LIVRE E SUSTENTABILIDADE

Uma vez apresentados alguns conceitos sobre software livre, tentaremos relacionar este tipo de software com sustentabilidade partindo da definição de SIQUEIRA [2010] que define sustentabilidade como uma conexão intrínseca entre justiça social, paz, democracia, autodeterminação e qualidade de vida.

Ao apresentarmos a tecnologia como um item de necessidade básico humano e verificarmos que tal item não se encontra disponível de forma simples, é necessário procurar uma forma de viabilizar para as pessoas o acesso à tecnologia da maneira financeiramente menos onerosa possível. A diminuição de

custos de hardware seria possível caso não houvessem patentes industriais envolvidas no processo de fabricação de um computador mas dificilmente é possível levar o desenvolvimento de peças a um custo zero mesmo depois do projeto estar pronto. Já o custo de software em um computador, que muitas vezes ultrapassa algumas vezes o custo de hardware, pode ser minimizado já que a cópia digital possui um custo muito baixo. Por esta razão, a inclusão digital e por sua consequência, a justiça social é mais facilmente alcançada se o software puder ser copiado sem custos, como o software livre. Aqui o acesso ao código-fonte pode ser necessário pois, caso o fabricante do software não tenha compilado o programa pra determinada plataforma, o acesso ao fonte permite ao usuário compilar. Certamente, se assumirmos a cópia não autorizada de software proprietário como uma opção para garantir o acesso das pessoas à tecnologia estamos não apenas infringindo a lei mas deixando de lado em nosso conceito de justiça o proprietário de tal software que deve ter respeitado o direito de não ser copiado.

Por meio da cópia digital autorizada, um recurso pode ser abundante e todos podem acessá-lo encerrando a luta por este recurso. Isto coloca um fim também na segregação social por meio do capital onde a regra de quem tem acesso ou quem não tem não é ditada pelo capital. Por esta razão, a cultura de paz também será mais facilmente alcançada por meio do software livre.

Além do acesso ao software propriamente dito, o software livre permite também o acesso ao seu desenvolvimento, por meio de suas comunidades. Nem sempre estas comunidades são democráticas, como é o caso das comunidades de software mantidas por grandes empresas. Mesmo sabendo que as decisões podem nem sempre serem tomadas por toda a comunidade é possível que parte das decisões que envolvem o futuro do software passe pela comunidade de usuários, tornando o software livre uma opção mais democrática que o software proprietário.

Quando uma comunidade não é democrática, o software desenvolvido por este projeto tende a ser duplicado por parte da comunidade que está insatisfeita e novos projetos surgem. Muitas vezes estes projetos combinam trechos de códigos e funcionalidades de projetos similares e isto cria uma teia de novas aplicações, similares com a ferramenta original mas distinta por alguma

característica. Tal teia de sistemas permite que o usuário tenha mais opções de ferramentas, as vezes similares, ou ainda a possibilidade de personalizar uma ferramenta para suas necessidades. Por isto, o software livre traz ao seu usuário a possibilidade de escolha e de decidir por si mesmo, também chamada de autodeterminação.

Por fim, mas não menos importante, o último elemento de sustentabilidade, a qualidade de vida, está implicitamente envolvida nos elementos anteriormente apresentados. Tais elementos permitem uma vida humana plena e em comunidade já que, nos dias atuais, tais comunidades podem estar distantes geograficamente mas muito próximas nas atividades diárias. Isto é possível desde que o acesso à tecnologia seja garantido.

4 CONCLUSÃO

Este artigo apresentou uma proposta de sustentabilidade tecnológica baseada nos princípios do software livre, defendendo o acesso à tecnologia como um item de necessidade básica humana. Boff [2012, pg 14] defende que sustentabilidade “fundamentalmente significa (...) o atendimento das necessidades da presente e das futuras gerações, e a continuidade, a expansão e a realização das potencialidades da civilização humana”. No caso da tecnologia, para expandirmos, acrescentarmos e realizarmos a continuidade do desenvolvimento, é necessário que a tecnologia seja distribuída e compartilhada, não apenas como produto mas como pensamento e código que possa ser alterado e adequado a cada necessidade.

Estamos o tempo todo cercados por sistemas computacionais e raramente nos questionamos sobre a continuidade e o desenvolvimento destas tecnologias. Este pensamento ingênuo pode ajudar a difundir ferramentas computacionais que criam a segregação das pessoas por meio do acesso à tecnologia mediado apenas por questões financeiras. Devemos evitar esta segregação e pensar em uma adoção tecnológica que seja sustentável e livre.

Os princípios e possibilidades do software livre foram confrontados com os conceitos de sustentabilidade apresentado por Siqueira [2010]. Há outras definições de sustentabilidade, não tão precisas, que incluem outros elementos nesta conexão como a preservação do meio ambiente, o pensamento complexo em detrimento do pensamento simplista e a cooperação. Tais elementos também

podem estar diretamente relacionados ao conceito de software livre e muito distante de software fechados e proprietários.

A cooperação, por exemplo, pode ser diretamente relacionada pois o desenvolvimento distribuído e colaborativo de uma aplicação depende diretamente do compartilhamento de seu código-fonte. Se não mostramos o que estamos fazendo, dificilmente conseguimos cooperar.

Já a economia de recursos que o compartilhamento permite não é certamente drástica mas pode ser significativa. Se as horas de gasto de energia elétrica usada para reimplementar um código que já foi implementado dezenas de vezes não nos toca como um recurso a ser economizado, certamente o desgaste humano gerado por tal tarefa não deve ser desprezado. Um código livre permite a “reciclagem de código”, a reutilização de um código que já foi usado, que possui experiência, que muitos olhos já olharam, que será menos suscetível à falhas e a erros, e que pode ser reusado por outros projetos.

Por fim, cito o pensamento complexo em detrimento do pensamento simplista ao nomear a educação como a grande possibilidade de mudança no comportamento do planeta. Se a educação passar pela educação tecnológica ela dependerá da existência de modelos e códigos que poderão ser lidos por estudantes para a criação do futuro. Estudantes estes que podem educados digitalmente com a ideia de que um código pode e deve ser compartilhado.

8 REFERÊNCIAS

BOFF, Leonardo. Sustentabilidade: o que é - o que não é. Petrópolis, RJ: Vozes, 2012.

BROCCHI, Davide The Cultural Dimension of Sustainability. In: S. Kagan, V.Kirchberg (Eds.), Sustainability: a new frontier for the arts and cultures, VASVerlagfür Akademische Schriften, Frankfurt am Main, 2008.

FLUSSER, VILÉM. O instrumento do fotógrafo ou o fotógrafo do instrumento?. 1982. Disponível em: <http://www.flusserbrasil.com/art31.html>. Acessado em 21/08/2017.

GATES, Bill et al. An open letter to hobbyists. Homebrew Computer Club Newsletter, v. 2, n. 1, p. 2, 1976.

GONZALEZ-BARAHONA, Jesus M. et al. Understanding how companies interact with free software communities. IEEE software, v. 30, n. 5, p. 38-45, 2013.

KROAH-HARTMAN, Greg; CORBET, Jonathan; MCPHERSON, Amanda. Linux kernel development: How fast it is going, who is doing it, what they are doing, and who is sponsoring it (an august 2009 update). Linux Foundation, 2009.

SEN, P., GAU, J., SEGALLER, S., CRINGELY, R. X., & CRINGELY, R. X. (1996). Triumph of the nerds. New York, NY, Ambrose Video.



1º Simpósio Internacional de Artes Urbanidades e Sustentabilidade
São João del-Rei, Brasil. 25, 26 e 27 de outubro de 2017

SIQUEIRA, Adilson R. Arte e sustentabilidade: argumentos para a pesquisa eco-poética da cena. *Moringa: teatro e dança*, João Pessoa, v. 1, n. 1, p. 87-99, jan. 2010. Disponível em <<http://periodicos.ufpb.br/index.php/moringa/article/view/4800>> Acesso em 5 jul. 2017.

STALLMAN, Richard. "The GNU operating system and the free software movement." (1999).