



Capítulo 03. Conceitos e discussão sobre software livre, software aberto e software proprietário

Grupo de estudos Licença Aberta – USP

Universidade de São Paulo – USP

(São Paulo, Brasil)

Neide Bueno

(Colaboradores: Alberto Steimber, Nely Mila B. Vati, Lila Red)

RESUMO

Este capítulo visa expor alguns conceitos e definições sobre o “Software Livre” e “Open Source Software” (OSS) e suas diferenças em relação ao uso do software proprietário bem como algumas vantagens da utilização do Software Livre ou Open Source Software.

OBJETIVOS DE COAPRENDIZAGEM

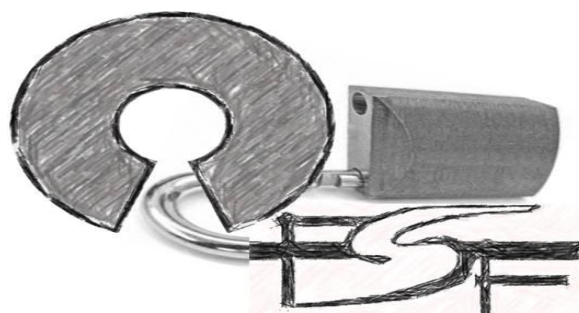
O objetivo direcionado para os leitores deste texto é refletir sobre estas diferenciações e implicações referentes aos Recursos Educacionais Abertos (REA).

POSSIBILIDADES DE REUTILIZAÇÃO

Este conteúdo pode ser reutilizado por todos os interessados em debater este tema.

PALAVRAS-CHAVE: REA, Software Livre, Open Source Software, software proprietário

1. ABERTURA



REA 01: Open Source Software, Proprietary Software and Free Software Foundation

Autores: Alberto Steimber

Fonte: [Wikimedia Commons](#)

Descrição: A figura acima foi gerada pelo autor com o aplicativo [Power Point](#) – mais detalhes pode ser acessado na Tool library

Objetivos: Explorar a relação entre Open Source Software, Proprietary Software and Free Software Foundation

Licença Aberta: CC BY SA

A imagem acima (REA 01) e' uma remixagem com os logos existentes dos conceitos-chave deste texto. Iniciamos a imagem com o logo de software de fonte aberta representado pela letra “O” de open com uma abertura.



Em seguida foi adicionada a representação de software proprietário através do cadeado que necessita de código de acesso para abrir e usar. Finalmente, foi incluído o logo do software livre, (representado pelas letras FSF – Free Software Foundation).

A intenção desta composição de logos e' refletir sobre as perguntas abaixo que são o ponto inicial para explorar o conteúdo deste capítulo.

- O que e' software livre?
- O que e' software de fonte aberta (opensource)?
- Quais as diferenças entre estes conceitos e software proprietário?
- Qual a relação entre estes conceitos e REA?

2. INTRODUÇÃO

Na era da tecnologia da informação, o desenvolvimento de um software tornou-se sinônimo de poder econômico e, em muitos casos, quase um monopólio através de um modelo de negócio concentrado em um determinado grupo ou empresa.

Por outro lado, a disseminação da tecnologia da informação, através de um sistema de desenvolvimento de software colaborativo em que o conhecimento é compartilhado – o chamado Software Livre ou Open Source – é possível se obter uma tecnologia mais estruturada e desenvolvida, contrapondo algumas vantagens em relação ao uso do software proprietário, mas não eliminando certos riscos.

Este capítulo visa esclarecer conceitos e definições sobre o “Software Livre” e “Open Source Software” (OSS) e principalmente, debater sobre as suas diferenças em relação ao uso do software proprietário bem como algumas vantagens da utilização do Software Livre ou Open Source Software.

A intenção e' oferecer maior esclarecimento sobre esta temática de software e direitos de uso para fazer melhores escolhas e tomadas de decisão na hora de optar por aplicativos para construir REA.

3. ORIGEM DO SOFTWARE LIVRE

Segundo PACITTI(2006), a metodologia altamente cooperativista defendida pelos adeptos do Software Livre, começou no fim dos anos 50, na forma de um modelo de software colaborativo ou cooperativo, cujos grandes beneficiados eram os fabricantes de hardware, que absorviam as contribuições do espírito cooperativo dos poucos usuários de uma época em que economicamente se dava mais valor ao hardware do que ao software em si.

O ilustre autor relembra que foi a partir dos anos 70, quando surgiram as “Software Houses Independentes”, iniciou-se a preocupação com a proteção e os direitos de propriedade intelectual e/ou empresarial do software. Dessa forma, instituíram-se as licenças, os direitos de autoria e propriedade, e outras restrições nos negócios comerciais. Naquela época – que então só era vendido o hardware – o software começou a ser comercializado de forma explícita e independente.

No final da década de 70, as empresas começaram a impor restrições aos usuários com o uso de contratos de licença de software. Em 1983, Richard Stallman iniciou o projeto GNU – com o objetivo de criar um sistema operacional totalmente livre, em que qualquer pessoa teria direito de usar, estudar, modificar e redistribuir o programa e seu código fonte, desde que garantindo os mesmos direitos para todos. Em 1985 Richard Stallman fundou a Free Software Foundation (FSF) e introduziu os conceitos de software livre e copyleft, os quais foram especificamente desenvolvidos para garantir que a liberdade dos usuários fosse preservada (Wikipedia, 2012).

O open source/código aberto deu relevância a uma nova prática de produção de software, só possível a partir dos anos 1990, com a disseminação da internet. A rede passou a funcionar como uma metáfora do mundo natural, em que os códigos mais competentes, melhores escritos e mais inovadores, encontravam programadores dispostos a aplicá-los e a melhorá-los. A relação mais flexível com a propriedade passou a ser justificada pela melhoria técnica, pela seleção natural estabelecida na internet.



Atualmente, com a disseminação da tecnologia da informação, tem-se um modelo de desenvolvimento de software colaborativo em que o conhecimento pode ser compartilhado por usuários em todo o mundo – o chamado Software Livre ou Open Source – através do direito de acessar os códigos fontes, efetuar modificações e redistribuí-los, obtendo-se uma tecnologia mais estruturada e desenvolvida. Alguns softwares livres conhecidos são o Linux, o ambiente gráfico KDE, o compilador GCC, o servidor web Apache, o OpenOffice.org e o navegador web Firefox, entre muitos outros.

4. SOFTWARE LIVRE X OPEN SOURCE

Para a maioria das pessoas – que não pertencem a comunidade do chamado “Software Livre” – existe uma grande dúvida se as expressões “Software Livre “ e “Open Source” traduzem o mesmo significado. O que se verifica é que na comunidade de usuários / desenvolvedores de software livre, alguns adotam a expressão “Software Livre” ou “Free Software”, enquanto outros utilizam “Código Aberto “ ou “Open Source”.

O que se sabe ao certo é que a expressão “free”, empregada pela comunidade, não significa “grátis”, mas está relacionada com a expressão “liberdade”. De acordo com a definição da Free Software Foundation (FSF, 2012), o “Software Livre” está vinculado à existência simultânea de 4 tipos de liberdades básicas para os usuários / desenvolvedores do software:

- a) A liberdade de executar o programa, para qualquer propósito ;
- b) 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;
- c) A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo ;
- d) 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.

Com relação a expressão “open source”, segundo a definição da open source initiative (osi, 2012) o nome adotado não significa somente ter acesso ao código fonte, mas o denominado “open-source software” deve seguir os seguintes critérios:

- 1) Livre Distribuição – A licença não deve restringir de nenhuma parte a venda ou distribuição do software gratuitamente, como um componente de distribuição de um software agregado contendo programas de diversos códigos diferentes. A licença não exige pagamento de royalty ou qualquer outro pagamento para essa venda.
- 2) Código fonte – O programa deve incluir seu código fonte e deve permitir a sua distribuição também na forma compilada. Se o produto não for distribuído com o código fonte, deverá haver uma forma bem divulgada de se obter o código fonte com um custo razoável para reprodução ou downloading via internet. O código deve ser legível e inteligível para que qualquer programador possa modificar o programa. Não é permitido que o software contenha um código fonte deliberadamente inacessível para um outro programador, ou formas intermediárias tais como “output” de um preprocessor ou tradutor.
- 3) Trabalhos Derivados – A licença deve permitir modificações e trabalhos derivados, e deve permitir que eles sejam distribuídos sobre os mesmos termos da licença do software original.
- 4) Integridade do autor do código fonte – A licença pode restringir o código fonte de ser distribuído em uma forma modificada apenas se a licença permitir a distribuição de arquivos patch (de atualização) com o código fonte para o propósito de modificar o programa no momento de sua construção. A licença deve explicitamente permitir a distribuição do programa construído a partir do código fonte modificado. Contudo, a licença pode ainda requerer que programas derivados tenham um nome ou número de versão diferentes do software original.
- 5) Não discriminação contra pessoas ou grupos – A licença não pode ser discriminatória contra qualquer pessoa ou grupo de pessoas.



- 6) Não discriminação contra áreas de atuação – A licença não deve restringir qualquer pessoa de usar o programa em um ramo específico de atuação. Por exemplo, ela não deve restringir o uso do programa em um negócio de empresa ou para pesquisa genética.
- 7) Distribuição da Licença – Os direitos associados ao programa devem ser aplicáveis para todos aqueles cujo o programa é redistribuído, sem a necessidade da execução de uma licença adicional para estas partes.
- 8) Licença não específica a um produto – Os direitos associados ao programa não devem depender que o programa seja parte de uma distribuição específica de software. Se o programa é extraído desta distribuição e usado ou distribuído dentro dos termos da licença do programa, todas as partes para quem o programa é redistribuído devem ter os mesmos direitos como aqueles que são concedidos junto com a distribuição do software original.
- 9) Licença não restrita a outros software – a licença não pode conter restrições a outros softwares que são distribuídos junto o software licenciado. por exemplo, a licença não pode insistir que todos os outros programas distribuídos sob o mesmo meio devam ser open-source software.
- 10) Licença neutra em relação a tecnologia – Nenhuma cláusula da licença pode estabelecer uma tecnologia individual ou estilo de interface a ser aplicada no programa.

Conforme as definições oficiais de Software Livre e de “Open Source Software” (OSS) , acima mencionadas, verifica-se que suas diferenças são sutis. A OSI define o código aberto usando a definição Debian de software livre, que é considerada pela comunidade apenas um detalhamento das 4 liberdades da FSF e que, dessa forma, todo software de código aberto é também um software livre (Wikipedia, 2012).

Porém, verifica-se uma discordância básica dos movimentos Software Livre e Código Aberto, sob o ponto de vista ideológico e comercial.

O Movimento do Software livre é considerado um movimento social que apregoa valores morais, éticos, direitos e liberdade, e que critica o uso e desenvolvimento de software proprietário por considerar que não é ético aprisionar conhecimento científico, que deve estar sempre disponível para permitir a evolução do conhecimento tecnológico.

Já o movimento pelo Código Aberto, é um movimento mais voltado ao mercado, conivente com o desenvolvimento de software proprietário, apregoando que o software aberto traz diversas vantagens técnicas e econômicas, levando as empresas a adotarem o modelo de desenvolvimento de software livre.

Como a diferença entre os movimentos “Software Livre” e “Código Aberto” está apenas na argumentação em prol dos mesmos softwares, é comum que esses grupos se unam em diversas situações ou que sejam citados de uma forma agregadora através da sigla “FLOSS” (Free/Libre and Open Source Software) (Wikipedia, 2012)

De fato, para a maioria da comunidade técnica, não há diferenças substanciais em que os termos software livre e código aberto pretendem definir. Ambos estabelecem praticamente os mesmos parâmetros que uma licença de software deve conter para ser considerada livre ou aberta. Ambas estabelecem, na prática, que o software deve respeitar aquelas quatro liberdades básicas que a FSF estabeleceu. Mas os defensores do termo “código aberto” afirmam que o termo fez com que os empresários percebessem que o software livre também pode ser comercializado (Cobertura Wiki, 2012).

Neste contexto, segundo uma das correntes interpretativas, o open source seria todo o programa de computador que se faz acompanhar de seu código-fonte e que não necessariamente precisa seguir todas as liberdades previstas pelo software livre. Open Source pode ser um programa que é distribuído com seu código-fonte , mas com uma licença que não permite sua livre distribuição, por exemplo. Outro exemplo são os programas que acompanham seu código-fonte, mas que só podem ser utilizados para fins não-comerciais.

Dessa forma, um Software Livre é um programa open source necessariamente, pois para que uma licença seja respeitada é preciso que o código fonte o acompanhe ou que esteja à disposição da Comunidade. Mas nem todo open source é um software livre (Cerqueira & Moniz, 2004).



Na realidade, se considera que o movimento Free Software e o movimento Open Source são como dois campos políticos dentro da comunidade de software livre.

É importante, também, diferenciar o Software Livre e o Open Source Software do software em domínio público. O primeiro, quando utilizado em combinação com licenças típicas (como as licenças GPL e BSD), garante a autoria do desenvolvedor ou organização. O segundo caso acontece quando se expira o prazo de proteção do software de acordo com a legislação de cada país. No Brasil, por exemplo, a proteção autoral de um software é de 50 (cinquenta) anos e sua proteção através do sistema de patente é feita somente em casos muito específicos, diferentemente dos Estados Unidos.

Mesmo que a proteção de um software tenha expirado, quer seja sob a legislação autoral ou de direitos de propriedade industrial, ainda assim, um software em domínio público pode ser considerado como um software livre, desde que atenda aos quatro tipos de liberdade para os usuários do software definidas pela Free Software Foundation, citados acima.

A liberdade de executar o programa significa a liberdade para qualquer tipo de pessoa física ou jurídica utilizar o software em quantas máquinas quiser, em qualquer tipo de sistema computacional, para qualquer tipo de trabalho ou atividade, sem nenhuma restrição imposta pelo fornecedor.

A liberdade de redistribuir o programa compilado, isto é em formato binário, necessariamente inclui a obrigatoriedade de disponibilizar seus códigos-fonte. Caso o software venha a ser modificado e o autor da modificação queira distribuí-lo, gratuitamente ou não, será também obrigatória a distribuição do código fonte das modificações, desde que elas venham a integrar o programa. Não é necessária a autorização do autor ou do distribuidor do software para que ele possa ser redistribuído, já que as licenças de software livre assim o permitem.

Para que seja possível estudar ou modificar o software – para uso particular ou para distribuir – é necessário ter acesso ao código-fonte. Por isso a disponibilidade desses arquivos é pré-requisito para a liberdade do software. Cada licença determina como será feito o fornecimento do código fonte para distribuições típicas, como é o caso de distribuições em mídia portátil somente com os códigos binários já finalizados (sem o fonte).

A maioria dos softwares livres é licenciada através de uma licença de software livre, como a GNU GPL, pela qual a fonte deve ser disponibilizada em local de onde possa ser acessado, ou deve ser entregue ao usuário, se solicitado, sem custos adicionais (exceto transporte e mídia).

5. SOFTWARE LIVRE E COPYLEFT

O Copyleft surgiu nos EUA e é derivado do instituto do copyright, do qual o desenvolvedor do software desde do primeiro licenciamento, libera os direitos de uso, reprodução, distribuição e alteração do código a qualquer interessado.

Segundo a Free Software Foundation (FSF, 2012), pelas regras do copyleft qualquer um que distribui o software, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o programa. O copyleft garante que todos os usuários tenham essa liberdade. Ou seja: se você recebeu um software com uma licença livre que incluía cláusulas de copyleft, e se optar por redistribuí-lo (modificado ou não), terá que mantê-lo com a mesma licença com que o recebeu.

Nem todas as licenças de software livre incluem a característica de copyleft. A licença GNU GPL (adotada pelo kernel Linux) é o maior exemplo de uma licença copyleft. Outras licenças livres, como a licença BSD ou a licença ASL (Apache Software License) não incluem a característica de copyleft.

Licenças como a GPL contêm um conceito adicional, conhecido como Copyleft, que se baseia na propagação dos direitos. Um software livre sem copyleft pode ser tornado não-livre por um usuário, caso assim o deseje. Já um software livre protegido por uma licença que ofereça copyleft, se distribuído, deverá ser sob a mesma licença, ou seja, repassando os direitos.

Associando os conceitos de copyleft e software livre, programas e serviços derivados de um código livre devem obrigatoriamente permanecer com uma licença livre, sendo que os detalhes de quais programas, quais serviços e quais licenças são definidos pela licença original do programa. O usuário, porém, permanece com a possibilidade de não distribuir o programa e manter as modificações ou serviços utilizados para si próprio.



As licenças de software livre permitem que eles sejam vendidos, mas estes em sua grande maioria estão disponíveis gratuitamente.

Uma vez que o comprador do software livre tem direito às quatro liberdades definidas pela FSF, ele poderia redistribuir este software gratuitamente ou mediante remuneração. As versões pagas geralmente são acompanhadas de algum tipo de serviço adicional, como direito a assistência técnica por determinado período e manuais, por exemplo. Muitas vezes comprar o software é mais vantajoso para o cliente final que não tem muita experiência em programação (Campos, 2011).

6. SOFTWARE PROPRIETÁRIO X SOFTWARE LIVRE

O que determina se um software é livre ou proprietário é o direito autoral que se aplica sob o seu licenciamento e distribuição. Ou seja, o desenvolvedor do software, que é o autor do programa, nos termos da legislação que protege o direito do autor, tem a faculdade de restringir o seu uso, reprodução, alterações e adaptações por parte de terceiros, ou autorizar / licenciar certos usos, de forma específica e sob remuneração (royalties). Nessa hipótese, estamos falando de software proprietário.

Software proprietário, portanto, são aqueles programas que não liberam os seus códigos fonte e que são comercializados como commodities. No modelo adotado pelo software proprietário seus usuários não tem acesso aos elementos técnicos que o compõem, tais como o código-fonte comentado, memorial descritivo, especificações funcionais internas, diagramas, fluxogramas e outros dados técnicos necessários à absorção da tecnologia.

No caso do software livre, onde o código é aberto e qualquer pessoa poderia ter acesso as informações técnicas sob o seu desenvolvimento, o usuário poderia efetuar as alterações e modificações necessárias no produto à medida que hardware se torna-se obsoleto. Dessa forma, o usuário não teria que pagar para fazer upgrade de programas, a cada versão nova do produto.

Nesse contexto, o software proprietário tende a concentrar seus usuários em torno da empresa fabricante como forma de aumentar seu lucro e sua participação do mercado. Por essa razão é conhecido também como “software monopolista”, mesmo que não haja efetivamente um monopólio. Exemplo notório dessa situação é a Microsoft, considerada a maior empresa de software do mundo, com seus programas inseridos em cerca de 90% dos computadores do planeta.

De fato, estudos realizados no início de 2000, por profissionais e estudiosos da área de TI sobre os riscos de segurança causados pela dominância dos produtos da Microsoft, como demonstrado em julgamentos nos Estados Unidos, adicionou complexos e “pesados” códigos ao seu sistema operacional não porque tal complexidade na programação era necessária, mas porque garante que fabricantes de computadores, usuários irão utilizar produtos daquela empresa ao invés de produtos dos concorrentes (Cerdeira e Moniz, 2004).

Ademais, os softwares proprietários têm sido construídos de uma forma que exige a constante troca de equipamentos ou hardware, o que gera uma dependência técnica. Por outro lado, os desenvolvedores e usuários do OSS sabem que o OSS produto não ficará obsoleto pela obsolescência da plataforma original do hardware, como acontece com os softwares proprietários que, nesse caso, acabam perdendo o atrativo comercial.

De fato, com a disseminação da tecnologia da informação, a grande questão no futuro é saber quem terá o controle sobre a tecnologia de software. Por um lado, temos aqueles que defendem um modelo centralizado e concentrado de desenvolvimento tecnológico (software proprietário) e por outro lado, temos aqueles que defendem um modelo de conhecimento compartilhado, onde qualquer pessoa pode entender os processos pelos quais a tecnologia é estruturada e desenvolvida (Software Livre ou Open Source).

A maioria dos projetos geralmente empregam mecanismos para ordenar todas as contribuições dos desenvolvedores do software de uma forma coerente, consistente e estável do produto. Dessa forma, o código adaptado para um produto OSS tem o mesmo padrão de um software produzido comercialmente, e com a disponibilidade do código aberto o produto pode ser adaptado para trabalhar com novos hardware, tonando-se um diferencial em relação aos software proprietário.



Ademais, constata-se que o produto desenvolvido pelo Software Livre ou Open Source é economicamente igual ou mais viável que o software proprietário pois representa um novo modelo de negócio. Ocorre que as empresas de software livre geram suas receitas com a prestação de serviços na área de consultoria, customização e desenvolvimento de aplicações específicas. Dessa forma, quanto mais o software for distribuído, gratuitamente, mais viável é o seu modelo de negócio.

A IBM, que não tinha o software como modelo de negócio, passou a utilizar o open source software e investir maciçamente no sistema operacional Linux. Os substanciais investimentos da IBM é um dos indicadores de que a natureza da competição na indústria de software tem mudado significativamente nos últimos 20 anos (Samuelson, 2006).

Mesmo entre empresas, cujo modelo de negócio é exclusivamente sobre software proprietário, o open source tem impactado competitivamente. A Microsoft atualmente tem utilizado produtos com acesso ao código. Após a Firefox introduzir inovações em seu browser , a Microsoft efetuou revisões no Internet Explorer. A Firefox fez um acordo com o Google para disponibilizar uma pesquisa da Google em seu browser, sendo que este projeto de open source tem uma fonte assegurada de receita e que pode ser usado em outros desenvolvimentos de open source, gerando novos modelos e negócios (Samuelson, 2006).

Por outro lado, existem um certo receio das empresas na utilização de Software Livre ou Open Source relacionada a segurança e a vulnerabilidade do sistema, considerando que sua natureza “livre” com a disponibilização de código pode facilitar sua adulteração por parte de hackers (Ford, 2007). Se for possível, para um técnico no assunto, descobrir a vulnerabilidade de um software, ou seja, certos pontos de “fraquezas” suscetíveis de serem explorados, este terceiro poderá desenvolver uma maneira explorar o sistema do software e promover um cyberattack e, prejudicar consideravelmente os negócios de uma empresa, gerados por aquele software.

7. ATIVIDADE DE COAPRENDIZAGEM: TOOL-LIBRARY

O ambiente “Tool-Library” – Biblioteca de Software para REA. criado no KMi Open University para o projeto europeu OpenScout, refere-se a um ambiente de rede social para integração de aplicativos (tools) e práticas (cenários) sobre web 2.0, e redes sociais interessantes para criação de REA na área educacional e administrativa. Este vídeo apresenta o projeto OpenScout-ToolLibrary



REA 02: Vídeoclip Educacional Aberto de Coletividades de Pesquisa para Publicação Internacional via Tool-Library

Autores: Colearn-AleOkada, Izabel Meister, Alexander Mikroyannidis, Suzanne Little

Fonte: [YOUTUBE](#)

Descrição: A figura acima foi gerada pelos autores com o aplicativo [iMovie](#) – mais detalhes pode ser acessado na Tool library

Objetivos: Conhecer o projeto OpenScout-ToolLibrary e projeto REA da Comunidade Colearn

Licença Aberta: CC BY SA

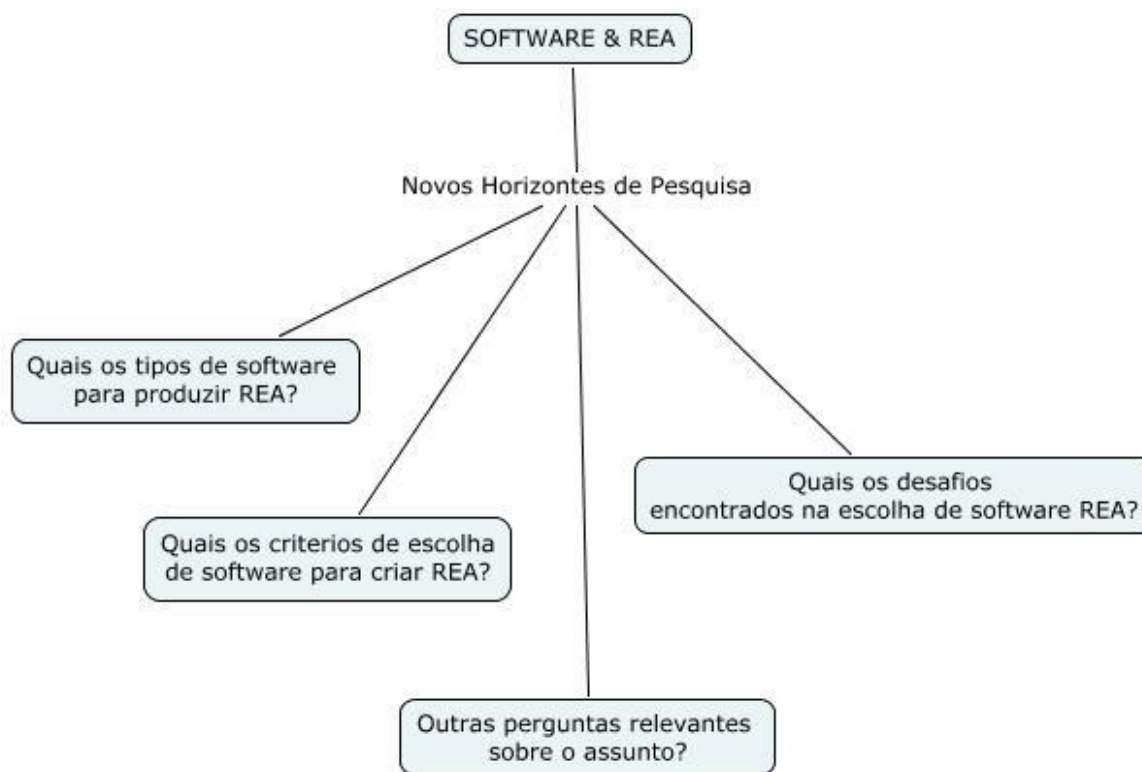
Os participantes da Comunidade Colearn estão debatendo varias temáticas importantes relacionadas aos conceitos apresentados neste texto. Os participantes do Colearn são grupos de pesquisa constituídos de professores e estudantes do Ensino Superior interessados no uso de tecnologias digitais na educação formal e informal.



Convidamos, então, todos os leitores a participarem do fórum de discussão aberto da Tool-Library e inclusive explorar o tipo de licenças apresentadas pelos softwares e aplicativos registrados na Tool-Library.

8. CONCLUSÃO E MAPA CONCEITUAL DE FUTURAS PESQUISAS

Nesse breve estudo refletimos sobre algumas diferenças entre os conceitos dos denominados Software Livre e Open Source e suas diferenças em relação a adoção do Software Proprietário. Ademais, que a adoção de um sistema de desenvolvimento de software em que o conhecimento é compartilhado – através do Software Livre ou Open Source – é possível se obter uma tecnologia mais estruturada e desenvolvida em relação ao software proprietário, inclusive relacionada a novos modelos de negócios como os serviços derivados da utilização de um código aberto. Entretanto, ainda existem receios por parte das empresas na utilização do Software Livre ou Open Source, principalmente relacionada as questões de segurança e vulnerabilidade que um sistema de Código aberto pode gerar e causar prejuízos nos negócios das empresas. Isso pode envolver também Instituições privadas de Ensino interessadas na produção de seus recursos educacionais (abertos ou não). Este tema apresenta várias questões (veja REA 03) para novos debates e convidamos os leitores para trazer novas indagações para ampliar e aprofundar este estudo.



REA. 03: Mapa de Futuras Pesquisas

Autores: Neide Bueno e Alexandra Okada

Fonte: [Wikimedia Commons](#)

Descrição: A figura acima foi gerada pelos autores no [Cmap Tools](#)

Objetivos: Conhecer e contribuir com novos horizontes de pesquisa

Licença Aberta: CC BY SA



REFERÊNCIAS

- Campos, A. (2011). O que é Software Livre. Disponível em: <http://br-linux.org/faq-softwarelivre/>. Acesso em 08/10/2011
- Cordeira, P. e Moniz, P. (2004). Cyberinsecurity: The Cost of Monopoly – How the Dominance of Microsoft’s Products Poses a Risk to Security, set. 2003. Citado no artigo Copyleft e Software Livre: Uma Opção pela Razão – Eficiências Tecnológica, Econômica e Social – II”. *Revista da ABPI* nº 71 – Jul/Ago 2004.
- _____(2004). “Copyleft e Software Livre: Uma Opção pela Razão – Eficiências Tecnológica, Econômica e Social – II”. *Revista da ABPI* nº 71 – Jul/Ago 2004.
- Cobertura Wiki (2012) Debate sobre Software Livre x Open Source. Disponível em: http://wiki.softwarelivre.org/bin/view/CoberturaWiki/Pol%edticaELinguagemNosDebatesSobreOSoftwareLivre#Software_Livre_x_Open_Source. Acesso em 13/10/2011.
- Ford, R.(2007) Open vs Closed . *ACM QUEUE* February 2007.
- FSF (2012) – Free Software Foundation. Disponível em: <http://www.fsf.org/about/>. Acesso em 13/10/2011
- Kemp, R. (2009). Current developments in Open Source Software. *Computer Law & Security Review* 25 (2009) 569-582
- OSI (2012) – Open Source Initiative. Disponível em: <http://www.opensource.org/>. Acesso em 13/10/2011
- Pacitti, T.(2006). *Paradigmas do Software Aberto*. Rio de Janeiro. LTC Ed. 119.p.
- Samuelson, P. (2006). IBM’s Pragmatic Embrace of Open Source. *Communications of the ACM*. October 2006/ Vol. 49. Nº 10.
- Watson, R. (2008) “The Business of Open Source”. *Communications of the ACM*. April 2008/Vol. 51. Nº 4
- Wikipedia (2012). Software Livre. Informações disponíveis em http://pt.wikipedia.org/wiki/Software_livre. Acesso em 08/10/2011.

CITAÇÃO

Bueno, N. (2012). Conceitos e discussão sobre software livre, software aberto e software proprietário. In: Okada, A. (Ed.) (2012) *Open Educational Resources and Social Networks: Co-Learning and Professional Development*. London: Scholio Educational Research & Publishing.

LICENÇA

Este capítulo tem licença Creative Commons ([CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/))