

## **Free/Libre and Open Source Software na Gestão da Informação: A liberdade no Mundo Proprietário**

José Hugo Leite Junior (Faculdade Campo Real) [j\\_hugo\\_l\\_j@yahoo.com.br](mailto:j_hugo_l_j@yahoo.com.br)

Álvaro José Argemiro da Silva (Faculdade Campo Real) [alvaro.argemiro@hotmail.com](mailto:alvaro.argemiro@hotmail.com)

### **Resumo:**

Cada vez mais presente nas organizações, o Free/Libre and Open Source Software (FLOSS) emerge nos sistemas das organizações, proporcionando em razão de suas características, aumento da competitividade no setor organizacional.

No que tange a competitividade os sistemas de informação despertam especial interesse, devido ao fato de que a maioria das organizações possuem suas operações muito parecidas, sendo que cada ganho que possa ser obtido, será de fundamental importância para a construção das estratégias e vantagens competitivas, permitindo assim a sustentabilidade por meio desse diferencial.

Sendo o sistema de Informação a rede neural da empresa, portanto, o responsável pelo feedback da maioria das atividades da organização, torna-se necessário então, discutir acerca do FLOSS nesses ambientes, levantando apontamentos que propiciem um melhor entendimento do modelo, para que, desta forma se estabeleçam delineamentos sobre a aplicação desses softwares nos ambientes de trabalho.

**Palavras chave:** Software Livre, Organizações, Gestão da Informação.

## **Free/Libre and Open Source Software in the informational management: the freedom in the world proprietary**

### **Abstract:**

Ever more common in the organizations, the FLOSS emerges in the organizations systems, providing because of its characteristics, the increase of competitiveness in the organizational sector.

In relation to competitiveness, the information systems arouse special interest, due to the fact that the majority of organizations have similar operations, being that, each gain that can be reached, is the fundamental importance for strategies constructions and competitive advantages, allowing the sustainability by means of this differential.

Since the information system is the neural net of the company, therefore, the responsible for the feedback of the most activities in the organization, it is needed to discuss about the FLOSS in these contexts, raising points that provide a better understanding of the model, and, in this way to establish outlines about the applying of these softwares in the workplaces.

**Key-words:** Free Software, Organizations, informational management

## 1. Introdução

O presente trabalho elenca alguns levantamentos bibliográficos acerca da possibilidade de pesquisa sobre o crescente uso da Tecnologia FLOSS (Free/Libre and Open Source Software) nas empresas tendo como prisma a viabilidade e o uso crescente dessa prática. Concentra-se em constatar o estado da arte deste modelo na gestão da informação, ao invés de propor uma implementação propriamente dita.

Considerando levantamentos iniciais, cabendo definir de pronto o que entendemos por Sistema de Informação, sendo que comungamos com a visão de Costa (2007), o qual explica:

“A informação é tudo na administração de uma organização. Mas para uma boa informação é necessário existir um conjunto de características para que esse fundamental instrumento de trabalho realmente atenda às necessidades dos gestores, como agilidade e confiabilidade. Entender como os Sistemas de Informação funcional são fundamentais para uma empresa colocar em prática aquilo que realmente precisa de um sistema”.

Nessa perspectiva, torna-se crucial em um primeiro momento, os delineamentos agregados a essa pesquisa, assim como nossos entendimentos.

## 2. Referencial Teórico

O sistema de informação torna-se imprescindível na demanda de informações em uma empresa, pois é esse sistema que irá definir sua qualidade, competência e competitividade no mercado, visto que é a informação que fornece o retorno à empresa, desde como ela está sendo vista pelos seus clientes ou como está ocorrendo o seu processo de produção. Entendemos aqui que o Sistema de Informação é a rede neural das organizações e que acompanhando Cidral, a efetividade dos sistemas de informação só é alcançada a partir de um processo de implementação que seja capaz de integrar as dimensões tecnológica, humana e organizacional desses sistemas (CIDRAL, 2005).

Segundo Laudon e Laudon *in* Cidral (2005), “Embora os sistemas de informação sejam a peça fundamental nas organizações, nem sempre a sua implementação tem alcançado o objetivo a que se propõe. Estatísticas mostram que cerca de 31% dos projetos de software são cancelados, 51% dos projetos que chegam ao fim custam duas a três vezes mais que o orçado e levam três vezes mais tempo do que o previsto e 75% dos grandes sistemas operam com falhas”. Nesse sentido, dentro de todo o planejamento de uma organização na instalação e uso de um Sistema de Informação, é necessário estudo, aplicações, convivências na empresa. Um Sistema de Informação não se trata de protocolos a serem seguidos, mas de procedimentos cotidianos das organizações. O que é preciso? Como vai ser feito? Que áreas ele agregará? É seguro? Qual o custo? Ficaremos mais eficientes e/ou eficazes? São questionamentos que devem ser feitos, pois, de nada adianta a uma organização ter um sistema sólido se não sabemos como usar, ou se ele não agrega valores qualitativos para uma empresa. Para Stair (1998), “estudos mostram que o envolvimento de administradores e tomadores de decisões

em todos os aspectos dos Sistemas de Informação é um fator fundamental para o sucesso da empresa, inclusive para aumentar os lucros e diminuir os custos”.

E para ser uma empresa com um potencial de qualidade, segundo Queiroz (1999), “O homem necessita de informações para gerar soluções inovadoras. Não basta existirem inovações consolidadas, é necessário que elas cheguem ao conhecimento de pessoas para serem utilizadas como base para o desenvolvimento de novas ideias”. Nesse sentido, os softwares livres permitem aos seus usuários e seus desenvolvedores compreenderem, estudar e estar inovando frequentemente o seu sistema, possibilitando agregação de novas tecnologias, interpretações e buscando assim uma melhora frequente no software.

Sendo assim, é perfil do Profissional de Engenharia de Produção, segundo Torkomain e Piekarski, *in* Batalha (2008) “decidir adotar uma nova filosofia de gestão que refletirá em mudanças culturais muito mais tecnológicas, embora o suporte tecnológico exista e seja necessário”. Portanto, a implantação de um Sistema de Informação parece prática, porém enfrenta uma grande barreira das tradições, quando agregamos a esse software o conceito de Open-Source ou de Software Livre. De acordo com Levy (1993), “O cúmulo da cegueira é atingido quando as antigas técnicas são declaradas culturais e impregnadas de valores, enquanto as novas são denunciadas como bárbaras e contrárias à vida.”. Open-Source já é uma realidade em nossos ambientes. Muitos de nós usamos o Firefox ou Google Chrome com navegador de internet, ou algumas distribuições de Sistemas Operacionais, como Mandriva, Ubuntu, Debian, entre outros. É fato que essa tecnologia permeia nossas práticas cotidianas e penetra cada vez mais em nossos ambientes, seja profissional ou pessoal, haja vista a corrida de consumidores para compra de Ipad, que possui em seu software, componentes de tecnologia Open-Source.

Como argumenta Cruz (2003), “Qualquer tecnologia da informação deve dar ao usuário o controle efetivo da informação, além de simplificar a operacionalidade de sua atividade; caso contrário, todos perdem.” Dentro dessa perspectiva de um conhecimento aberto a todo o grupo responsável por essa operação, temos como enfoque a expectativa de desenvolvimento de Softwares seguindo princípios do Open-Source.

Se perguntássemos há uns 10 anos atrás, poucas pessoas saberiam o que era o sistema Operacional Linux, muitos usuários ainda não sabem usá-lo. Fato é, que esse sistema tem permeado o ambiente de organizações, e isso, claro que não se deve à ideologias de desenvolvedores de equipes ligadas ao ramo de Sistemas de Informação, mas há algumas vantagens competitivas, que esse sistema traz para a empresa que o agrega ao seu ambiente.

Cruz (2003) aponta alguns requisitos para a Tecnologia de informação:

“Tecnologia ultrapassada é tecnologia cara!

Tecnologia desatualizada é tecnologia que atrapalha!

Tecnologia desconhecida causa perda de oportunidade!

Tecnologia desequilibrada causa prejuízo!”.

Procuraremos assim, enquadrar alguns parâmetros do FLOSS com os princípios de Cruz. O sistema Linux com o uso de seus Repositórios permite que o usuário fique sempre atualizado com seus programas. Os repositórios permitem que sejam realizadas atualizações frequentes que funcionam a partir do momento em que localiza-se um Bug, erro de software, os repositórios permitem que sejam corrigidos os agravantes do sistema, arrumando ao mesmo tempo os erros de software, gerando uma nova compilação e atualizando o próprio sistema.

Como o sistema Linux parte do princípio Open-Source, ele garante que o conhecimento do software não fique agregado apenas a uma pequena equipe de desenvolvedores, mas a uma comunidade global que trabalha 24h/dia, seguindo assim o princípio de Raymond.

Quem já teve a oportunidade de usar um sistema Operacional Linux, principalmente nos quatro últimos anos, período em que esse Software teve um importante desenvolvimento no Brasil; poderá dizer que raramente viu seu sistema invadido por vírus e precisou formatá-lo, o que é comum nem Sistemas Operacionais proprietários. O Linux garante uma maior segurança de transmissão e armazenamento de dados, fator importante para um sistema de informação, por ser um software que dificilmente pode ser invadido. Não que isso seja impossível, porém a proteção desse sistema é maior que a do seu concorrente proprietário, uma vez que os quatro níveis de importância de um usuário para o sistema, permitem que nem todos os usuários consigam fazer alterações ou instalar programas, além de que os vírus que circulam pela rede praticamente a maioria são destinados para o sistema operacional mais popular.

Contudo, para melhor compreendermos a abordagem sobre FLOSS, partimos agora para uma breve discussão acerca do conceito e da funcionalidade desse modelo de processo de desenvolvimento de Software.

### 3. Histórico do FLOSS

Neste item será feito um breve histórico do Software Livre baseado nos autores Ascenso e Santos (2003), Hauben (2003), Lottermann (2001), Stallman (1996) e Tanenbaum (1994).

Nas décadas de 50 e 60 o software era produzido nos meios acadêmicos em que trabalhavam em cooperativismo. Esses sistemas eram distribuídos e mantidos pelas comunidades de usuários. O código fonte nesse momento era distribuído devido a necessidade de consertar bugs ou adequar melhor o funcionamento do Software.

Durante a criação da Arpanet3 (*Advanced Research Projects Agency Network*), os desenvolvedores dos Laboratórios Bell e da American Telephone and Telegraph (AT&T) sentiram a necessidade de um sistema para controlar os diversos processos e recursos “in-house computer” controlados por eles (ASCENSO e SANTOS, 2003), resolvendo assim, criar o sistema operacional BESYS (*Bell Operating System* – Sistema Operacional da Bell) para ser utilizado como seu computador. Entretanto, o sistema foi requisitado por outras pessoas e a Bell começou a fornecer o software de maneira gratuita, porém, sem suporte técnico.

Em 1964, os Laboratórios Bell tiveram que decidir se usariam um sistema desenvolvido por terceiros ou se criariam um novo sistema operacional. Após reunir-se internamente a empresa optou pela segunda opção, criando assim o Multics (Multiplexed Information and Computing Service), unindo forças com a General Electric e com o Massachusetts Institute of Technology (MIT).

Em 1969 a AT&T retira-se do projeto, entendendo que o desenvolvimento do Multics aconteceria em um longo prazo e com custos altíssimos. Ken Thompson, Dennis Ritchie e outras pessoas, resolvem então, escrever uma versão simplificada do Multics. Foi implementado assim o sistema de arquivos e um interpretador de comandos, digitados pelos usuários (*Shell*). O sistema atendia às necessidades quanto à realização de trabalhos de desenvolvimento. Posteriormente é chamado de Uniplexed Information and Computing Service (Unics), uma sátira ao sistema operacional Multics e posteriormente muda-se para Unix (HAUBEN, 2003).

Em 1970, Dennis Ritchie inicia a linguagem C, que foi assim usada para reescrever o Unix. Logo então em 1974, Thompson e Ritchie publicam um artigo sobre o Unix, incentivando assim muitas universidades a copiar o software da AT&T. Contudo, como na época a empresa era apenas destinada à área de telecomunicações, não atuaria na área da informática, o que ocasionou o oferecimento sem restrições de uso do programa nas universidades, que poderiam desta forma estudar e reestruturá-lo o programa da melhor maneira que conviesse.

Nesse momento iniciam-se os encontros para discutir sobre o Unix, emergindo ideias e adaptações que foram absorvidas pelo sistema rapidamente. Isso só aconteceu porque na época, o compartilhamento de programas e seus códigos-fonte era comum, permitindo assim que se desenvolvessem estudos para melhoria do software, fazendo com que surgissem várias versões do Unix.

É na década de 1980 que a comunicação ganha destaque, cria-se o protocolo de comunicação o TCP/IP (Transmission Control Protocol/Internet Protocol – Protocolo de Controle de Transmissão/Protocolo Internet), que foi usado pelo departamento de sistemas da universidade de Berkeley para manter a comunicação, mesmo a distância, entre os Unixes, este procoloto torna-se então, o padrão na Internet.

Devido a sua praticidade o Unix começa a ser amplamente utilizado, sendo apoiado por empresas e instituições as quais contribuíam para desenvolvê-lo. Porém, em 1984 a AT&T, é dividida em várias companhias independentes, emerge assim uma sucursal destinada à informática, ocasionando o fechamento do Unix. É importante mencionar, que neste período os computadores pessoais já eram populares, contudo portadores de softwares proprietários nos sistemas.

Neste mesmo ano, Stallmann, decide criar um sistema operacional em que qualquer pessoa tenha privilégios e possa copiá-lo, usá-lo, alterá-lo e distribuí-lo. Ele queria ter uma comunidade de desenvolvedores trabalhando juntos novamente, sendo que com a perda dos softwares abertos, sua comunidade foi praticamente extinta. Para isso acontecer era imprescindível que o software tivesse o seu código-fonte aberto (LOTTERMANN, 2001). Segundo Stallman, era necessário desenvolver um software que os programadores pudessem, a partir de outro existente, melhorá-lo em seu benefício, sem a necessidade de desenvolver outro sistema. Inicia então o projeto GNU (fsf.org).

Stallman sabia que para facilitar a migração dos usuários para o novo programa seria necessário que este fosse compatível com o Unix, era uma das primeiras estratégias para o desenvolvimento do projeto GNU (Gnu Is Not Unix – Gnu não é Unix). No entanto, como a licença Copyriht mantém os direitos autorais, ele desenvolve a Copyleft (a esquerda do Autor), “*Copyleft pola sua vez está formado por duas palavras: copy (copia) e left que pode ser traduzido tanto por esquerdo como polo particípio do verbo leave (deixar, abandonar), de jeito que poderíamos traduzir copyleft por: deixai, copiar.*”(FLORES, 1998). Cria-se assim, uma nova licença, permitindo às pessoas copiarem, executarem, alterarem, modificarem e distribuírem o programa, desde que esse continue com as mesmas opções. A fim de viabilizar recursos, o projeto GNU integrou programas livres, mesmo que não fossem enquadrados nos critérios regidos pela licença GNU. O MIT fornece então sua estrutura à Stalmann, para que este desenvolva o sistema GNU. Os primeiros programas foram o compilador gcc (Gnu C Compiler) e o editor de textos Emacs.

No ano de 1985 ele funda a FSF (Free Software Foundation – Fundação Software Livre) para desenvolver e manter sistema o GNU, que recebia o seu faturamento através de venda de cd-roms de Software Livre. Primeiramente desenvolvem o interpretador de comandos BASH (Bourne Again Shell). O nome é um acrônimo para a (Bourne-Again Shell), um trocadilho com Stephen Bourne, o autor do ancestral direto do atual Unix shell sh , que apareceu em the

Bell, Sétima Edição, versão Research Labs do Unix (GNU.org., 2009) e aprimoram assim diversos programas, que poderiam ser rodados em Unixes proprietários, mesmo sem existir o sistema GNU completo.

No início da década de 1990 o sistema GNU estava praticamente pronto, porém necessitava desenvolver o kernel (núcleo) do sistema, parte do sistema que mantém o diálogo software/Hardware. O núcleo que está sendo desenvolvido pelo GNU foi chamado de HURD, sofreu vários atrasos, mas outros componentes foram concluídos em 1991. O GNU Debugger e Emacs foram sucesso.

Em 1991, Linus Torvalds, da Universidade de Helsinki, na Finlândia, desenvolveu um núcleo compatível com o Unix (utilizando o gcc), baseado no Minix, um sistema operacional desenvolvido para fins educacionais. Linus logo em seguida disponibiliza o código-fonte na Internet para um melhor aprimoramento do programa. Em 1992, o kernel Linux foi agregado ao GNU, completando assim o Sistema Operacional, que passou a se chamar GNU/Linux, embora atualmente seja conhecido apenas como Linux.

Outro destaque do mundo do software livre emerge na década de 1990, Eric Raymond, que através da sua publicação “A Catedral e o Bazar”, influencia a Netscape a liberar o acesso ao código-fonte do seu Navegador, hoje conhecido como Mozilla Firefox. Raymond, a Netscape e outros, passaram a buscar alternativas para aproximar a indústria de software desenvolvido para o comércio seguindo os princípios do Open-Source. É o início dos princípios que rege as diretrizes do modelo FLOSS.

Frente à isso, com o aumento do tamanho de projetos Open-Source surgiram repositórios e sites centralizadores de softwares, entre eles: The Apache Group, Mozilla.org, Openoffice.org, Linux Online. Nestes sites, os programadores podem transferir os códigos-fonte, comunicar erros e falhas, apresentar correções e novos recursos, trocar experiências e realizar documentação.

Atualmente, existem várias empresas que investem em programas baseados no Linux com suporte comercial, comprometidas com os princípios do FLOSS. Nota-se nesse sentido um alto desenvolvimento desse modelo de software, e que cada vez mais empresas buscam neles suprir necessidades que antes não eram percebidas.

#### **4. Como Funciona o FLOSS**

Um dos principais trabalhos sobre o movimento Open-Source, é lançado por Eric Steven Raymond em 1997, em que apresentou no Linux Kongress (Evento realizado em 27 de Maio de 1997, em Würzburg, na Alemanha), um manifesto intitulado A Catedral e o Bazar, publicado como parte do livro com o mesmo nome em 1999, sendo considerado por muitos como o manifesto do movimento Open Source. Raymond nessa obra se consagrou com a frase: “Havendo olhos suficientes, todos os erros são óbvios”- inspirado por Linus Torvalds, desenvolvedor do Kernel do Sistema Operacional LINUX.

Entretanto, para darmos continuidade à discussão, torna-se necessário antes elencar alguns apontamentos que nos proporcionam uma maior compreensão entre a diferença de Software livre e Open-Source. Seguiremos essa discussão acerca da publicação encontrada no site da SUSE CORPORATION:

“O Software Livre caracteriza-se pela oferta de 4 liberdades básicas:

- a) A liberdade de *usar o programa*, para qualquer propósito (liberdade 0);

- b) A liberdade de *estudar como o programa trabalha*, podendo adaptá-lo às necessidades próprias (liberdade 1). Acesso ao código fonte é pré-condição para tanto;
- c) A liberdade de *redistribuir cópias*, para que você possa ajudar ao seu próximo (liberdade 2);
- d) A liberdade de *melhorar o programa*, e lançar suas melhorias para o público em geral, para que assim toda a comunidade se beneficie (liberdade 3). Acesso ao código fonte é pré-condição para isto.

O termo Código Aberto, ou Open-Source, em inglês, foi criado pela OSI (Open Source Initiative).

(...)

O movimento pelo **Código Aberto** foi iniciado no final do século XX, tendo se originado como parte de uma campanha de marketing para o Software Livre. Salienta os benefícios técnicos e econômicos do código-fonte aberto e livre ao desenvolvimento, e pouco ou nada sobre os aspectos éticos.

Como as diferenças entre esses movimentos, do **Software Livre** e do **Código Aberto**, são mínimas, centrando-se apenas na argumentação em favor dos mesmos softwares, amiudadamente os dois grupos se unem em diversas situações ou são mencionados, agregadamente, pela sigla "FLOSS" (Free/Libre and Open Source Software)".

Embora existam várias discussões sobre a política do Open Source, não apenas a concepção de Raymond e Stallman, o software possui aspectos diferentes dos projetos convencionais, (modelo de software proprietário) apresentando características bem rigorosas por parte da comunidade, dentre elas:

**Participação Voluntária:** Para um grande número de desenvolvedores é o risco e a tensão, os desafios de elaborar e estudar o software, aprimorar o que é inviável. O aprendizado fala mais alto e acaba sendo compartilhado para com toda a comunidade.

**Flexibilidade na escolha da função:** Um colaborador pode escolher qualquer projeto para ajudar a desenvolver, porém pelo bom senso, na maioria das vezes, colaboradores iniciantes são designados para exercer funções básicas, mas nada impede que um usuário mais experiente entre na comunidade gerenciando um projeto ou realizando programação avançada.

**Horário indeterminado:** Um dos grandes impasses da rotina do desenvolvimento do software com essa distinção, mas, que devido ao grande número de colaboradores não remete a uma perda significativa do processo de desenvolvimento, por atender a fuso-horários e culturas diferentes. O colaborador é quem realiza e define sua jornada de trabalho; como alguns colaboradores já possuem outra renda fixa, colaboram em um horário flexível ao seu.

**Planejamento:** Não existe uma sala onde todos os membros desenvolvedores discutem sobre o software, mas, diversos colaboradores em diferentes países, o que torna quase que impossível um planejamento das atividades.

**Atualização de Versões:** Não tendo em seu projeto de desenvolvimento um planejamento como nos moldes de softwares proprietários, por ter um trabalho voluntário e sem compromisso com prazo ou normas contratuais, o software é lançado quando a equipe determina que já é uma versão funcional; corrigir bugs eventuais e realizar melhorias, ocorrem após o uso e a análise dos usuários.

**Usuário colaborador:** usuários que fazem uso do software podem sugerir melhorias e ideias para os desenvolvedores, podendo até mesmo tornarem-se colaboradores.

**Acesso:** As distros, “é um sistema operacional Unix-like incluindo o kernel Linux e outros softwares de aplicação, formando um conjunto. Distribuições (ou “distros”) mantidas por organizações comerciais, como a Red Hat, Ubuntu, SUSE e Mandriva, bem como projetos comunitários como Debian e Gentoo montam e testam seus conjuntos de software antes de disponibilizá-los ao público” (CAMPOS, 2006), são feitas através da internet ou qualquer mídia de armazenamento.

**Brainstorms:** Enquanto o trabalho é desenvolvido e posteriormente quando já está finalizado, o software é armazenado nos seus repositórios, locais digitais de armazenamento dos softwares, esses projetos são praticamente realizados via fóruns de participação com acesso aos repositórios, o que facilita o acesso ao conhecimento e as discussões sobre os mesmos e assim, a comunidade pode realizar downloads e os desenvolvedores podem corrigir os erros.

## **5. Quanto da implementação do Software Livre**

O exército Brasileiro começou em 2000 a migração dos seus sistemas para Software Livre "Trata-se de uma poderosa ferramenta de trabalho e de livre distribuição, com custo nulo" (Revista Linux, 2001). Nesse processo de migração destacaram alguns fatores-chaves para a aplicação deste tipo de software em seus sistemas. Entendemos então, que estes fatores aplicam-se como base também para a implementação em empresas privadas, devido aos pressupostos aqui apresentados:

### **5.1 Economia de custos a médio e longo prazo com software fechado**

Como é do conhecimento de praticamente todo o setor que precisa de Sistemas de informação para sustentar a prática cotidiana da empresa, são altos os custos de licenças de uso de softwares para o uso nas empresas, alguns softwares freeware (Software adquirido através do pagamento de licença para o uso), por exemplo, permitem a sua licença para o uso residencial, porém, para o uso privado existe a obrigatoriedade da compra da licença.

Outro fator crítico para o uso de software livre é a alta necessidade de manutenção proporcionada pelo Software Proprietário, pois geralmente esses softwares, rotineiramente sofrem travamento, erros de sistema, ou armazenam algum programa malicioso que deixa o sistema lento, necessitando obviamente assim, de um serviço de manutenção; sendo que quando a empresa, geralmente de pequeno porte, não tem uma central de desenvolvimento de informática, acaba tendo que requisitar serviços de terceiros. Se uma empresa possui um volume interessante de estações de trabalho dez, por exemplo, esses custos acabam gerando um grande problema para ela, somente no que se refere à manutenção de softwares.

Problema constante é a abrangência do software, já que quanto mais usuários uma empresa possui, menor será o feedback de informações e mais demorado será para aplicar as alterações necessárias às organizações. Existem softwares genéricos, que atendem aos princípios básicos, mas não à total realidade da empresa. Se o problema do Software é um conflito de Hardware/Software, a questão se torna maior, pois é necessário para o desenvolvedor e sua equipe analisar o problema, enquanto que para o Open-Source, temos vários desenvolvedores para acurar o problema.

### **5.2 Segurança, estabilidade e disponibilidade, proporcionadas pelo FLOSS**

Tendo em vista que o mercado atual está focado em relações via internet, tornam-se necessários alguns critérios-chaves para o que está sendo exposto nesse ambiente. Pois, a internet não apenas facilita a comunicação entre as empresas, mas também qualifica empresas à expor seus serviços e seus produtos para um público maior.

Segundo Daniel Biaggio, administrador de rede, Sylvio Iura e Marcos Luvizan, gerentes de tecnologia da Toyota: “o Linux tem trazido vantagens ao TI da empresa Economia foi outro



ponto atraente do Linux para a Toyota. Além da possibilidade de o sistema ser utilizado em plataforma Intel, a estabilidade diminui a necessidade de manutenção, o que reduz os gastos, inclusive, com pessoal. Com isso, é necessário apenas uma pessoa para gerenciar os servidores E-mail/Web/DNS da Toyota do Brasil. “A relação custo/benefício é excelente, visto que temos o sistema funcionando sem nenhum tipo de problema”, afirma Biaggio. Os hardwares utilizados são servidores Pentium III de 900MHZ, com 512MB RAM e 30GB SCSI”. (Revista Linux, 2001)

Contudo, não apenas na exposição dos seus produtos, mas o simples fato de a empresa estar conectada à internet significa que está vulnerável para receber em seus Discos Rígidos e nas suas mídias de armazenamento, programas maliciosos, que com softwares proprietários tornam a empresa mais vulnerável a esses contínuos ataques. É necessário assim, integrar uma política de Software livre de forma a garantir a acessibilidade.

### **5.3 Descontinuidade de suporte a versões**

Outro fator importante para o uso de software livre são as inesperadas atualizações dos softwares proprietários, atualizações constantes, garantem lucros maiores aos desenvolvedores, mas não às empresas, as empresas desenvolvedoras, acabam pensando na empresa, e não nos seus clientes, diferentemente de um software proprietário, que é desenvolvido para atender as demandas dos usuários, podendo adequar facilmente o software aos seus requisitos.

Muitos desenvolvedores de Softwares livres mantêm alguns serviços de atualizações e manutenções nos sistemas pagos. Porém, é um serviço que necessariamente não precisa ser feito pelos desenvolvedores, pode ser feito por qualquer profissional da área, gerando custos ou não, garantindo assim uma maior liberdade ao usuário do programa.

Geralmente quando ficamos presos a um software, também ficamos presos a vontade de seus desenvolvedores, um exemplo seria a descontinuidade da versão de Software, e o abandono de atualizações e manutenção do serviço prestado. Obrigando praticamente o usuário a vincular-se a versão mais recente, pois precisa desta para que a sua empresa funcione corretamente. Porém no software livre se é lançada uma versão mais nova nos repositórios o usuário já pode adquirir sem se preocupar com custos para mudar de versão.

### **5.4 Independência tecnológica**

Um dos mais interessantes fatores para o uso de um Software Livre é a independência tecnológica que esse software proporciona aos seus usuários. Os usuários, sejam eles residenciais ou comerciais, não dependem dos fabricantes e seus custos absurdos para aprimorar o software para o seu uso, podendo ainda usá-lo em quantas máquinas forem necessárias, e também distribuí-lo para quem assim o desejar. Ou seja, o usuário torna-se independente do desenvolvedor. Seguindo o site da Suse, “O Software Livre e de Código Aberto também é uma maneira pela qual, empresas podem compartilhar custos de desenvolvimento. Por exemplo, Novell e Red Hat são concorrentes, mas ambas desenvolvem muitos dos mesmos programas e, assim, ajudam uma a outra. IBM e HP também podem ser vistas como concorrentes embora ambas contribuam para o kernel Linux, etc., partilhando assim os custos de desenvolvimento”.

### **5.5 Independência de um único fornecedor**

Perante este fator podemos argumentar os baixos custos para manter o sistema e realizar a sua manutenção, como já explanado anteriormente, mas isso se deve ao fato de a empresa possuir quando usuária de software livre, uma independência monopolística, diferentemente dos usuários do Sistema Operacional Windows, em que tem a dependência inerente aos interesses

da Microsoft para desenvolver várias vertentes de uma mesma versão de Software comumente, como no caso do Windows 7 chamados de (Starter, Home, Profissional, Ultimate). Pois se somente ela é detentora dos direitos autorais do programa somente ela tem os direito de modificar o software. Porém, no caso do sistema operacional Linux, a empresa verifica qual a distribuição desse sistema operacional mais interessa ao seu perfil empreendedor, como praticidade, estabilidade, segurança, alguns fatores cruciais e importantes quando se refere a qualidade na Gestão dos Sistemas de Informação. O Software Livre torna possível um mercado para suporte competitivo, aumentando potencialmente a qualidade do suporte. Já com o software proprietário, na realidade, apenas o detentor dos direitos, que tem acesso ao código-fonte, pode oferecer suporte decente; tendo, portanto, uma espécie de monopólio.

### **5.6 Possibilidade de auditabilidade dos sistemas**

Diferente de um software proprietário, os softwares livres permitem aos administradores de rede saberem muito bem o que esta operando nas máquinas da sua empresa. No sistema Windows, por exemplo, líder do mercado, observamos que alguns softwares podem roubar dados importantes como senha de bancos, números de Documentos, etc. O caso do Linux é mais simples, enquanto no Sistema Windows temos milhares de programas mal intencionados a solta na rede, para o Linux o número total não chega a centenas. Como podemos confiar em um sistema que não sabemos os processos que estão sendo realizado internamente nos softwares.

Para a Suse, “Com software proprietário você não podes saber o que um determinado programa realmente faz. Alguns bem conhecidos de software proprietário foram pegos espionando usuários e enviando informações sobre seu comportamento e tal. Software proprietário também tem uma tendência a incluir várias restrições digitais sobre o quê o usuário pode fazer, quando, por quanto tempo, etc. Com Software Livre tens acesso ao código-fonte e pode estudar o quê e como o programa faz e alterá-lo se você não gostou do que ou momo ele faz”.

“O software de Código Aberto permite que mais pessoas vejam o código-fonte e que assim o corrija, ele pode se desenvolver mais rapidamente e se tornar melhor. Este sistema de "peer review" (revisão por pares) pode ser comparada à maneira como funciona a pesquisa científica. Em comparação com o código proprietário que mantém em segredo código-fonte, que raramente pode ser visto por alguém estranho a empresa ou por trás dele.”  
(pt.opensuse.org)

### **5.7 A Grande ameaça de vírus, trojans e worms e ameaças de segurança que usuários tem no sistema proprietário, é praticamente inexistente no FLOSS.**

Em consulta ao site da Avira (<http://www.avira.com/pt-br/support-virus-lab>), encontramos uma lista que demonstra 6494 tipos de programas mal intencionados como Vírus, Trojans, Worms, Malware, Exploits, Backdoors, etc. Dentre os tipos de programas mal intencionados apenas os Backdoors atingem diretamente o Linux, enquanto todos da lista atingem o Windows e com uma frequência relativamente maior. Este fato acontece na maioria das vezes, devido a má configuração da senha de root ( administrador máximo do software) no Sistema Operacional Livre com um baixo nível de segurança, o que faz com que o sistema fique vulnerável à maiores invasões.

## **6. Conclusão**

O mundo do FLOSS ainda é recente nos ambientes organizacionais, raros são os livros em bibliotecas destinadas ao tema, foi necessário abordar uma bibliografia em sua maioria virtual, devido a essa dificuldade de acesso ao livro físico.

Acerca dos apontamentos aqui levantados, procuramos demonstrar que o FLOSS cresce cada vez mais em ambientes empresariais que buscam competitividade e rompem com os tradicionalismos cotidianos pensando em uma melhora tecnológica para sua organização.

Outro fator relevante que leva a adoção deste modelo de tecnologia em software está relacionado ao fato da redução dos custos e consequentemente a possibilidade de sua adoção cada vez maior por empresas e/ou entes de governo que procuram agir dentro da legalidade, evitando dessa forma o software pirata e ao mesmo tempo aumentando a eficiência das ações.

Essa pesquisa deverá ter continuidade visto que, este é o início para uma abordagem amostral sobre a presença dos Softwares livres nas organizações.

## 7. Referências

**ASCENSO, M. F., SANTOS, P. M. de S.** *História e desenvolvimento do Sistema Operativo Unix*. Disponível em: <<http://www.eq.uc.pt/~pedrosa3/Unix1.htm>>. Acesso em 03 de setembro de 2011.

**AUDY, J. L. N.; ANDRADE, G. K. de; CIDRAL, A.** *Fundamentos de Sistemas de Informação*. Porto Alegre. Bookmark, 2005.

**BATALHA, M. O.** *Introdução à engenharia de produção*. Rio de Janeiro: Elsevier, 2008.

**CAMPOS, A.** *O que é uma distribuição Linux*. BR. Florianópolis, março de 2006. Disponível em <<http://br-linux.org/faq-distribuicao>>. Consultado em 20 de julho de 2011.

**COSTA, C. E.** *Sistemas de Informação – Sistema de gestão empresarial*. Disponível em: <<http://www.administradores.com.br/informe-se/producao-academica/sistemas-de-informacao-sistemas-de-gestao-empresarial/358/>> em 14/07/2010.

**CRUZ, T.** *Sistemas de informações gerenciais: tecnologias da informação e a empresa do século XXI*. Atlas, 3. ed. – São Paulo, 2003.

**FLORES, R.** *O que é Copyleft*. Disponível em: <<http://ramonflores.br.tripod.com/GNU/copyleft.html>>. Acesso em 20 de outubro de 2010.

**Free Software Foundation.** *What is free software and why is it so important for society?* Disponível em: <<http://www.fsf.org/about/what-is-free-software>> acessado em 3 de janeiro de 2011.

**LÉVY, P.** *As tecnologias da Inteligência – O futuro do pensamento na era da informática*. São Paulo. Editora 34. Tradução de Carlos Irineu da Costa. 2004

**LOTTERMANN, J.** *Viabilidade da Utilização de Software Livre em um Ambiente Empresarial*. Dissertação de Mestrado. Puc-Campinas, 2001.

**MINISTÉRIO DA DEFESA.** Plano de Migração para Software Livre no Exército Brasileiro. Disponível em: <[http://www.softwarelivre.gov.br/clientes/softwarelivre/softwarelivre/casos/Plano\\_Migracao\\_Soft\\_Livre\\_13FEV07.pdf](http://www.softwarelivre.gov.br/clientes/softwarelivre/softwarelivre/casos/Plano_Migracao_Soft_Livre_13FEV07.pdf)> Acesso em 25 de Outubro de 2010.

**SILVA, F. J. S.** *Software Livre: Conceitos, História e Impactos*. Disponível em: <<http://www.deinf.ufma.br/~fssilva/palestras/2009/sl.pdf>> Acesso em 20/10/2010.

**UNIXSUPPORT.** *Histórias de Unix*. Disponível em: <<http://www.unixsup.com/unixlinux/historiaunix.html>>. Acesso em 21 de setembro de 2010.

**QUEIROZ, A. H.** *EMPATIA E INOVAÇÃO: Uma proposta de metodologia para concepção de novos produtos*. Dissertação de Mestrado. UFSC- Florianópolis. 1999

**REVISTA DO LINUX.** *ESTUDO DE CASO – A Toyota do Brasil adota (e adora) o pingüim. O motivo? O sistema é altamente confiável nas missões críticas*. nº 29, maio de 2002. Disponível em: <[http://augustocampos.net/revista-do-linux/029/caso\\_toyota.html](http://augustocampos.net/revista-do-linux/029/caso_toyota.html)>. Acesso em 12 de outubro de 2010.

**REVISTA DO LINUX.** *ESTUDO DE CASO – Braço forte, mão amiga: Eliminar barreiras no aprendizado de ferramentas livres foi o principal obstáculo para o Exército.* nº 23, novembro de 2001. Disponível em: <<http://augustocampos.net/revista-do-linux/023/caso.html>>. Acesso em 15 de outubro de 2010.

**STALLMAN, Richard.** *Projeto Gnu.* Disponível em: <<http://pt.scribd.com/doc/60125044/O-Projeto-GNU-Richard-Stallman>>. Acesso em 20 de Outubro de 2010.

**STAR, Ralph M.** *Sistemas de Informação: Uma abordagem Gerencial.* Segunda edição. LTC- Livros Técnicos e Científicos S.A. Rio de Janeiro, RJ, 1998.

**TANENBAUM, Andrew S.** *Sistemas Operacionais Modernos.* Rio de Janeiro: Livros técnicos e Científicos Editora S.A., 1999.

*Linux é 74% mais estável que Windows, aponta teste.* Disponível em: <<http://informatica.terra.com.br/interna/0,,OI210362-EI553,00.html>>. Acesso em 30 de novembro de 2010.

*Software Livre e de Código Aberto.* Disponível em: <[http://pt.opensuse.org/Software\\_Livre\\_e\\_de\\_C%C3%B3digo\\_Aberto](http://pt.opensuse.org/Software_Livre_e_de_C%C3%B3digo_Aberto)>. Acesso em 02 de janeiro de 2010

*Bash Reference Manual.* Disponível em: <<http://www.gnu.org/software/bash/manual/bashref.html>> . Acesso em 5 de novembro de 2010.