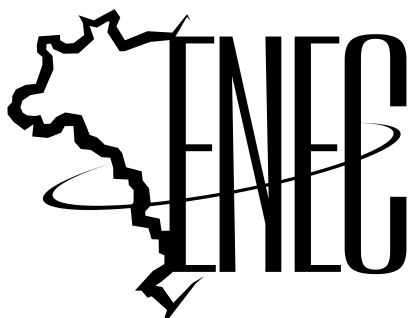


Cartilha de Software Livre



Executiva Nacional dos Estudantes de Computação

www.enec.org.br



projeto**software**livre
B A H I A

www.psl-ba.softwarelivre.org

3^a Edição
Junho de 2005

Sumário

Sobre a Cartilha	4
Versão modificada da ENEC	4
Nota de direitos autorais	5
Versão original do PSL-BA	5
Nota de direitos autorais	6
Software Livre	7
Histórico	7
Conceito de Software Livre	15
Licenças de Software	18
A Comunidade Software Livre	22
Por Que Usar Software Livre?	24
GNU/Linux	26
Sobre o GNU/Linux	26
Distribuições GNU/Linux	28
Interfaces Gráficas ou Gerenciadores de Janelas	32
Tabela de Programas Equivalentes	37

Inclusão Digital	39
O que é Inclusão Digital?	39
Divagações sobre Inclusão Digital	40
Tabuleiros Digitais	42
Acessibilidade	43
Links	46
Mais sobre Software Livre	46
Projeto Software Livre nos estados brasileiros	47
Distribuições GNU/Linux	49
Interfaces Gráficas (Gerenciadores de Janelas)	50
Alguns Softwares Livres bastante utilizados (conheça e use!)	51
Realização	53
Sobre o Projeto Software Livre Bahia	53
Sobre a ENEC	55
Próximos eventos de software livre	61

Sobre a Cartilha



Versão modificada da ENEC

Este documento é uma adaptação da Cartilha de Software Livre do Projeto Software Livre Bahia. Foram incluídos nesta, textos relacionados à ENEC - Executiva Nacional dos Estudantes de Computação, responsável pela distribuição destas cartilhas em eventos onde a ENEC estiver presente.

O item seguinte reproduz a citação original de direitos autorais da Cartilha de Software Livre do Projeto Software Livre Bahia.



Nota de direitos autorais

©2005, Executiva Nacional dos Estudantes de Computação.

Alguns direitos reservados. Este trabalho está licenciado sob uma Licença Creative Commons “Atribuição-Uso não Comercial-Compartilhamento pela mesma Licença”. Para ver uma cópia desta licença, visite [\\creativecommons.org/licenses/by-nc-sa/2.0/br/](http://creativecommons.org/licenses/by-nc-sa/2.0/br/) ou envie uma carta para Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

O conteúdo deste trabalho pode ser utilizado, copiado, distribuído ou modificado de acordo com os termos da sua licença de distribuição.



Versão original do PSL-BA

A Cartilha de Software Livre do Projeto Software Livre Bahia tem o objetivo de ser uma introdução simples, clara e objetiva aos principais conceitos do mundo do Software Livre.

Esta Cartilha é uma obra coletiva. As seguintes pessoas contribuíram na sua construção: Abelmon Bastos, Antonio Terceiro, Aurélio Heckert, Carla Elaine Freitas, Carla Schwingel, Charles Santana, Daniel Batista, Flávio Civatti,

Krishnamurti Nunes, Leandro Santos, Mônica Paz, Nelson Pretto, Paulo César Oliveira, Pedro Kröger, Thiago Tavares, Tiago Vaz, Vinícius Pinheiro, Wagner Dantas.

Esta cartilha é um documento livre. Você pode utilizar o seu conteúdo segundo os termos na licença Creative Commons “Atribuição-Uso não Comercial-Compartilhamento pela mesma Licença”.

Essa publicação foi feita inteiramente usando Software Livre. Os textos foram editados colaborativamente usando o TWiki, e a versão impressa foi feita usando ferramentas como o \LaTeX , `html2latex` e Inkscape.



Nota de direitos autorais

©2004-2005, pelos autores citados acima.

Alguns direitos reservados. Este trabalho está licenciado sob uma Licença Creative Commons “Atribuição-Uso não Comercial-Compartilhamento pela mesma Licença”. Para ver uma cópia desta licença, visite creativecommons.org/licenses/by-nc-sa/2.0/br/ ou envie uma carta para Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

O conteúdo deste trabalho pode ser utilizado, copiado, distribuído ou modificado de acordo com os termos da sua licença de distribuição.

Software Livre



Histórico

O que é um software? Como surgiu o Software Livre? O que o diferencia do software proprietário? Como o Software Livre evoluiu até os dias de hoje? Por que os movimentos pela disseminação do Software Livre crescem tanto em nosso país? Por que encontramos nele a solução para tantos problemas?

Este breve histórico não pretende responder a todas essas perguntas em tão pouco espaço. Esforça-se, contudo, em ser o pontapé inicial que dará impulso aos novos navegantes de um universo ainda pouco difundido, criado sobre uma perspectiva que preza pela colaboração e pela liberdade da informação. Sejam bem-vindos!

□ O Software

Software é o nome dado a qualquer programa de computador.

Ao contrário do hardware (monitores, impressoras, mouses, placas, memórias etc) o software não é algo físico e por isso não sofre desgaste ao longo do tempo.

Um software é, portanto, uma estrutura lógica, um programa que realiza funções dentro de um sistema computacional. E é, geralmente, desenvolvido por programadores que utilizam linguagens de programação para construí-lo. Softwares correspondem aos sistemas operacionais (Windows, Linux, Mac OS etc), drivers que controlam o comportamento de alguns hardwares (driver de modem, de impressora, de placa de vídeo etc) e todos os aplicativos utilizados pelos usuários finais, como editores de texto (Winword, Notepad, Gedit, OpenOffice.org Writer), planilhas eletrônicas (Excel, OpenOffice.org Impress, Gnumeric), navegadores internet (Mozilla, Internet Explorer, Opera), processadores de imagem (Gimp, Paint, CorelDraw, Adobe PhotoPaint), dentre outros.

O software, por não ser físico e sim lógico, pode ser duplicado e armazenado em disquetes, cds, discos rígidos (HD). Sua cópia pode ser transportada de um computador para o outro, desde que estejam conectados em rede.

□ Como surgiu o Software Livre?

Para entendermos o surgimento do Software Livre, é válido compreendermos, primeiro, o que é o software proprietário e o papel das grandes empresas de software na monopolização do conhecimento.

Como mencionado, um software é uma estrutura lógica desenvolvida por programadores. Essa estrutura lógica, isto é, os **tijolos** que compõem um software, correspondem aos bits: **0's e 1's** . Porém, um programador não pode desenvolver um programa utilizando-se de 0's e 1's, que é uma linguagem somente compreensível pela máquina — e não por seres humanos. Para tanto, os programadores utilizam linguagens de programação que possuem palavras chaves e estruturas que permitem enxergar um programa como uma receita de bolo, isto é, um conjunto ordenado de instruções, denominado **código fonte** . Por exemplo:

1. Programa pede que o usuário digite sua idade e tecla **Enter**
2. Programa recebe a informação (idade)
3. Se idade for maior ou igual a 18 mostra mensagem na tela: "Você já é um adulto!"
4. Se idade for menor que 18 mostra mensagem na tela: "Você ainda é um adolescente"

O código fonte seria a representação desses passos em uma linguagem de programação (C, C++, Perl, Python,

Pascal etc). Esse **código fonte** , ao ser processado por um outro programa, denominado compilador, transforma-se em 0's e 1's para que possa ser entendido pela máquina. Quem quiser modificar o programa precisa ter o código fonte, já que 0's e 1's não são compreendidos pelo homem.

Antigamente, os programadores compartilhavam seus códigos fontes uns com os outros e, assim, todos podiam modificar o programa e também partilhavam as mudanças. Este hábito era bastante difundido nas grandes universidades estadunidenses das décadas de 60 e 70 e sempre foi bastante condizente com o espírito acadêmico, cujos princípios de liberdade e cooperação se assemelham aos da **cultura hacker** .

□ O Software Proprietário

Quando o computador se mostrou viável como produto de distribuição massiva, as coisas começaram a mudar de rumo. As pesquisas em desenvolvimento de novas soluções cresceram, e muitas empresas adotaram como estratégia comercial a não-divulgação dos códigos-fonte dos programas.

As empresas vendiam seus softwares, mas não disponibilizavam seus códigos-fonte. O usuário só recebia o programa na linguagem de máquina (0's e 1's), o que tornava possível a sua utilização, mas não mais possibilitava o estudo e a modificação. Isso era estrategicamente interessante para as empresas: com o conhecimento sobre o software restrito a elas, cresciam as barreiras à entrada de novas empresas, menores, no mercado.

Diversos artifícios foram utilizados para manter restrito o conhecimento nos softwares: os programadores dessas empresas assinavam termos de compromisso de não divulgação dos segredos da programação; e os softwares vendidos possuíam licenças que impunham diversas restrições: além de impossibilitados de modificar o programa, não poderiam fazer cópias dos programas que eles adquiriam, e muito menos distribuir essas cópias. Alguns programas também só poderiam ser utilizados para fins específicos. Resumindo, o cliente não tinha controle sobre o software executado em seu equipamento. Ele podia somente comprar uma licença de uso, que permitia o uso daquele programa em apenas um computador.

Como estratégia para garantia dessas vantagens advindas da restrição do uso do conhecimento humano, a indústria de software proprietário associou o ato de compartilhar programas de computador ao de abordar navios, matar seus tripulantes e saquear a sua carga: a cópia não autorizada de software proprietário foi denominada de **pirataria** .

É importante ressaltar que o software distribuído pelas empresas contém, embutido, o conhecimento de milhares de programadores brilhantes, que por sua vez se valeram do conhecimento de várias pessoas que vieram antes deles, que desenvolveram teorias e conhecimento deixados para a humanidade. Estas empresas agora se apropriam de todo esse conhecimento acumulado, que poderia ser compartilhado com outras pessoas.

□ A Reação: Uma Licença para a Liberdade

Tudo permaneceu assim até que um grupo de **hackers** programadores do MIT (Instituto de Tecnologia de Massachusetts, nos EUA) perceberam que podiam fazer a diferença.



Os hackers, diferentemente de como são conhecidos e tratados pela mídia, são pessoas com princípios éticos, defensores da cooperação e da disseminação do conhecimento através da liberdade da informação. São caracterizados como pessoas de elevado conhecimento técnico, que amam o que fazem, e que sentem prazer em compartilhar seu saber com os outros. Os hackers, ao contrário dos *crackers* e dos *defacers*, não são criminosos digitais e possuem uma estrutura hierárquica implícita baseada na cultura da dádiva (*Gift Culture*), e definida apenas pelo mérito. Quanto mais contribuições um hacker oferece para a sua comunidade, mais *status* ele possui dentro dela. Os hackers também são chamados de *geeks*.

Essa reação começou pequena, com poucos colaboradores. Um dos programadores do MIT, Richard Stallman, enfurecido pela comercialização do conhecimento, fundou o Projeto GNU visando criar uma plataforma de software totalmente livre. Ou seja, no GNU qualquer pessoa poderia:

1. Utilizar o software para qualquer fim
2. Estudar o código do software
3. Modificar o código do software

4. Redistribuir cópias do software

Esses quatro direitos ficaram conhecidos como as quatro liberdades do Software Livre. Baseado nelas, o Projeto GNU redigiu uma nova licença de uso chamada GNU GPL (*GNU General Public Licence*). Dessa forma, todo software distribuído com tal licença estava “livre”. De acordo com a GNU GPL, a única condição para que alguém pudesse usufruir desses direitos era que passasse para as outras pessoas os mesmos direitos dos quais usufruiu. Vale lembrar que esta licença não proíbe a venda do Software Livre, somente garante que conterà as quatro liberdades mencionadas e que estas nunca poderão ser retiradas do software. Dessa forma, o software já nasce livre e permanecerá livre.

O projeto GNU também foi o responsável pelo desenvolvimento de inúmeros softwares livres. Para cada software proprietário existente, os desenvolvedores buscavam construir um outro similar, livre. Um dos softwares mais essenciais foi proposto pelo finlandês Linus Torvalds, que por conta própria desenvolveu o núcleo (*kernel*) de um sistema operacional, batizado de Linux. O Linux e os softwares desenvolvidos pelo Projeto GNU formaram o primeiro sistema computacional livre, o GNU/Linux, que atualmente é a principal alternativa ao Microsoft Windows.

Como o Software Livre não é mantido por uma entidade única, não pode ser comprado pela Microsoft ou por qualquer outra empresa. O Software Livre também não vai à falência, já que a única premissa para sua existência é a de que existam pessoas com necessidades e com disposição de compartilhar seus códigos-fonte. O número de indivíduos que participam dessa iniciativa e o número de

empresas que têm apostado suas fichas nessas idéias indicam que o software livre está em crescimento acelerado.

A GNU GPL não é a única licença de Software Livre. Existem muitas outras, e basta que elas garantam as quatro liberdades para que o software seja livre. Algumas licenças são menos restritivas que a GPL, não impondo, por exemplo, que trabalhos derivados devam ser distribuídos pelos mesmos termos. A escolha da licença depende de que características o desenvolvedor deseja atribuir à sua obra.

□ O Nosso Contexto

O Software Livre surge, então, da necessidade de abandonarmos o velho papel de meros usuários da tecnologia e passarmos a desenvolvê-la e usá-la para o bem de todos. O Brasil, particularmente, precisa acabar com a sua dependência tecnológica e passar a desenvolver softwares, ao invés de continuar refém dos preços abusivos impostos pelas grandes corporações e seus mercados.

O Software Livre é a nossa chance de tornar a tecnologia nossa aliada no desenvolvimento nacional. O Governo, sensível a essa oportunidade, vem incentivando cada vez mais o uso de Softwares Livres nas repartições públicas, reduzindo drasticamente os custos com licenças de software proprietário. Sendo que a economia deste recurso poderá ser redirecionada para investimentos em tecnologia nacional; ou até mesmo para setores mais problemáticos, como a saúde e a educação, minimizando a injustiça social.



Conceito de Software Livre

☐ Software Livre é Uma Questão de Liberdade, Não de Preço.

O termo Software Livre se refere à liberdade que o usuário tem de **executar, distribuir, modificar e repassar as alterações** sem, para isso, ter que pedir permissão ao autor do programa.

Pode ser definido mais claramente pelas quatro liberdades defendidas pela Free Software Foundation para os usuários de software:

- ☞ A liberdade de executar o programa, para qualquer propósito;
- ☞ 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;
- ☞ A liberdade de redistribuir cópias de modo que você possa beneficiar o próximo;
- ☞ 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.

Um programa será considerado livre se todos os seus usuários tiverem essas quatro liberdades.

Notem que os quatro itens acima não fazem nenhuma referência a custos ou preços. O fato de se cobrar ou não pela distribuição ou de a licença de uso do software ser ou ser não gratuita não implica diretamente o software ser livre ou não.

Nada impede que uma cópia adquirida por alguém seja revendida, tenha sido modificada ou não por esta pessoa.

Nada impede, também, que as alterações feitas num software para uso próprio sejam mantidas em segredo. **Ninguém é obrigado a liberar suas modificações**, se não quiser. **Porém, se escolher fazê-lo, é obrigado a distribuir de maneira livre**. Essa é uma observação importante a se fazer, porque muitas pessoas (especialmente corporações) têm receio de usar software livre porque temem que seus “concorrentes” tenham acesso a informações e métodos de trabalho privados. As personalizações não têm que ser distribuídas. A restrição é que, se elas forem distribuídas de alguma maneira, têm que manter as quatro liberdades descritas acima.

A liberdade de utilizar um programa significa a liberdade para qualquer tipo de pessoa, física ou jurídica, utilizar o software em qualquer tipo de sistema computacional, para qualquer tipo de trabalho ou atividade, sem que seja necessário comunicar ao desenvolvedor ou a qualquer outra entidade em especial. A liberdade de redistribuir deve incluir a possibilidade de se repassar tanto os códigos fontes quanto os arquivos binários gerados da compilação desses códigos, quando isso é possível, seja o programa original ou uma versão modificada. Não se pode exigir autorização do autor ou do distribuidor do software para

que ele possa ser redistribuído.

Para que seja possível modificar o software (para uso particular ou para distribuir), é necessário ter o código fonte. Por isso, o acesso aos fontes é pré-requisito para esta liberdade. Caso ele não seja distribuído junto com os executáveis, deve ser disponibilizado em local de onde possa ser copiado, ou deve ser entregue ao usuário, se solicitado.

Para que essas liberdades sejam reais, elas têm que ser irrevogáveis. Caso o desenvolvedor do software tenha o poder de revogar a licença, o software não é livre.

□ Diferenças entre Software Livre e Proprietário

“No software proprietário, o programador abdica da liberdade de controlar sua obra, em troca de salário e compromisso de sigilo. O distribuidor, fantasiado de ‘fabricante’, torna-se proprietário de tudo. Desde o código fonte, tido como segredo de negócio, até as cópias executáveis, licenciadas ao usuário sob custódia e regime draconiano. Enquanto no software livre o programador abdica de um dos canais de receita pelo seu trabalho, em troca da preservação do controle dos termos de uso da sua obra. Em contrapartida, se a obra tiver qualidades, agregará eficiência aos empreendimentos em torno dela. Seu valor semiológico, conversível em receita com serviços, será proporcional à magnitude do esforço colaborativo onde se insere. O código fonte é livre sob licença que preserva esta liberdade, enquanto a cópia executável é tida como propriedade do usuário. (...) Só tem a perder com ele (Software Livre) quem consegue galgar posições monopolistas no modelo proprietário. O problema é que a ganância faz

muitos acreditarem que serão os eleitos pelo deus mercado, enquanto seguem correndo atrás da cenoura amarrada na ponta da vara que pende das suas carroças digitais, não se importando com os efeitos colaterais de se tratar conhecimento como bem escasso, ao considerarem software como mercadoria.”

Texto de **Pedro Antonio Dourado de Rezende** publicado no Observatório da Imprensa



Licenças de Software

Para falar sobre Software Livre é indispensável que comecemos falando em Direitos Autorais e *Copyright*, uma expressão criada pelos estadunidenses com o objetivo de dar exclusividade de edição de materiais de imprensa escrita aos seus detentores.

Dessa forma, autores que possuíssem o *Copyright* de suas obras poderiam designar quem poderia, e como poderia, copiar e distribuir cópias de seus livros, artigos ou revistas. Entretanto, como não é necessário ser autor da obra para deter o seu *Copyright*, **não podemos dizer** que direito autoral é igual a *Copyright*.

Copyright pode ser definido dessa maneira:

“O único direito legal concedido a um escritor, a um editor, a um compositor, ou a um distribuidor para

publicar, produzir, vender, ou distribuir um trabalho artístico.”

E na legislação Estadunidense (pode ser visto em www.copyright.gov), na lei que define Copyright, está o texto abaixo (tradução dos editores):

“Usado para promover o progresso da ciência e das artes úteis, fixando-se por épocas limitadas aos autores e aos inventores o direito exclusivo sobre suas respectivas escritas e descobertas.”

Enquanto isso, o **Direito Autoral** é o direito que o autor do software tem de deixar seu produto sob um *Copyright* (veja que o *copyright* pode não ser seu), sem necessariamente estabelecer regras de uso, cópia e distribuição. Por exemplo, um funcionário faz um software em uma empresa e concorda em ceder-lhe o seu *copyright*, ele abre mão de definir como será usado, copiado e distribuído o programa, mas não deixa de ser o autor e pode reclamar se o software for usado para um fim não previamente acordado com a Empresa.

O sistema de *copyright* atribui **donos** ao software, e esses, pela definição apresentada acima, têm o direito de estabelecer regras de distribuição da obra. Devido a esse poder, surgiu o conceito de Software Proprietário, um paradigma de distribuição de software vigente desde o início da década de 80, que foi responsável pela criação do chamado **software de caixinha**.

Porém, ao contrário do que muitos pensam, simplesmente existir o *copyright* de um software não basta para

que ele seja Software Proprietário. O copyright, ou Direito de Uso, permite que o detentor desse direito estabeleça regras para o uso e distribuição do software. Essas regras são chamadas de **licenças de uso**, e podem estabelecer, por exemplo, se o software pode ou não ser distribuído gratuitamente, se pode ou não ser copiado, entre outras coisas...

Software Livre não é necessariamente de domínio público!, apesar de essa ser uma interpretação compreensível do conceito de **liberdade**, para quem ainda não se familiarizou com a quebra do paradigma do software proprietário.

Um programa de domínio público é aquele em que o criador abre mão de seus direitos de autoria e de licenciamento de cópias. Nesse caso, quem estiver de posse do código tem o direito de fazer dele o que desejar, sem ter que obedecer a qualquer restrição ou norma.

No paradigma do Software Livre, o autor do software resguarda seus direitos de criador, mantendo livres o uso e o conhecimento (do código fonte) do software para quem deles necessitar através da redação adequada de um *copyright*.

Dessa subversão do sentido do *copyright* para **proibir que haja restrições**, em vez de restringir o uso, vem o termo *CopyLeft*. Um trocadilho com a palavra em inglês (deixe copiar, ao invés de direito de cópia), mas que não tem significação legal.

Ao invés de encararmos o software como uma idéia, uma forma de se solucionar um problema, nós ainda o vemos como um item final de uma linha de produção; um

artigo físico, que se desenvolveu como qualquer outro produto, em escala industrial, e pelo qual se deve pagar. Nessa definição, o software se assemelha mais a uma cadeira ou a uma mesa, do que a uma fórmula matemática, ou a uma especificação científica.

Imaginem se o o grego Pitágoras tivesse podido, há séculos atrás, restringir a forma como deveríamos utilizar suas descobertas relacionadas a cálculos com círculos e triângulos; Todo o desenvolvimento matemático, de engenharia e arquitetura atuais estariam comprometidos.

Coisa semelhante acontece com programas de computador. Se considerarmos o programa como uma maneira de se resolver determinado problema, com a ajuda do computador, então fica mais simples aceitar o software como uma idéia, não como um bem.

E o que queremos é que as boas idéias sejam utilizadas em benefício de todos, e que todos possam usufruir das idéias e das inovações da tecnologia e da ciência.

Para garantir essa liberdade, no caso do software para computador, a Free Software Foundation redigiu algumas licenças, que, aplicadas ao programa, mantêm os direitos de autoria ao implementador do software, dando aos usuários do programa certas liberdades.

No caso da mais conhecida das licenças para Software Livre, a *GNU General Public License (GPL)*, essas liberdades incluem o direito de estudar, alterar para se adequar aos seus interesses, copiar e redistribuir o software, desde que nunca se subtraia do próximo usuário esses mesmos direitos.

Há outras licenças, algumas mais, outras menos restritivas. A *Lesser GPL (LGPL)* — também redigida pela FSF — por exemplo, é mais permissiva que a GPL. Bibliotecas de funções distribuídas sob a LGPL podem ser utilizadas por software proprietário, apesar de serem livres. Se essas mesmas bibliotecas estiverem sob a GPL, elas podem ser usadas apenas por programas também sob a GPL.



A Comunidade Software Livre

A Comunidade Software Livre Mundial hoje envolve mais de dez milhões de pessoas. Seu alicerce encontra-se na Free Software Foundation, e no projeto GNU, proposto por Richard Stallman, que ao resolver rebelar-se contra o fato de não se poder alterar o código de um software proprietário, buscou expandir a prática que considera ideal para o desenvolvimento de um produto tecnológico: a de que diferentes programadores podem ler o código, alterar, modificar, acrescentar, resolver problemas, propor outras soluções e até mesmo novas funções para um determinado programa. Assim, os membros da Comunidade Software Livre consideram o código de um sistema computacional semelhante ao conhecimento matemático: um ferramenta para se chegar a algo concreto.

A Comunidade de Software Livre é muitas vezes considerada como somente composta por *hackers*, devido ao fato de seus participantes possuírem um grande conhecimento tecnológico, mas pessoas das mais diferentes áreas e interesses (Direito, Comunicação, Administração) passa-

ram a contribuir para com os movimentos pelo Software Livre, devido a seus ideais libertários e lógica diferenciada de produção.

Por Que Usar Software Livre?

Para uma melhor compreensão dos motivos que cercam os usuários de Software Livre, vale lembrar a forma como ele surgiu e o que representa para a sociedade no contexto atual. O Software Livre surgiu baseado no conceito de **liberdade**, no qual as pessoas têm o direito garantido às quatro liberdades já mencionadas. Na visão filosófica do Software Livre, a liberdade não é um direito individual, mas um **direito coletivo** e por isso deve ser mantido e passado de pessoa para pessoa. Além disso, a premissa de qualquer projeto de Software Livre é a **colaboração** entre as pessoas interessadas, sem concentração de poder ou qualquer outro artifício que venha a ferir as liberdades já mencionadas.

Tendo em vista esse cenário, podemos facilmente citar algumas das razões para se utilizar Software Livre:

- ☞ Poder utilizar o software para qualquer finalidade;
- ☞ Ter acesso ao código fonte e poder modificá-lo, sem quaisquer restrições;

- ☞ Poder copiá-lo e executá-lo em quantas máquinas de-sejar;
- ☞ Poder distribuí-lo, sem violar, é claro, essas liberdades a que todos têm direito;
- ☞ Ter o seu computador equipado com software de qualidade a um custo baixo ou nulo;
- ☞ Não ficar preso às restrições impostas pelas licenças de softwares proprietários;
- ☞ Não ficar dependente de novas versões com preços abusivos que eventualmente apresentam incompatibilidades com versões antigas;
- ☞ Não ficar dependente de um fornecedor;
- ☞ Ficar livre da pirataria;
- ☞ Incentivar o desenvolvimento de tecnologia local;
- ☞ Interagir e compartilhar soluções com sua comunidade, seja física ou virtual;
- ☞ Lutar contra o monopólio de grandes corporações que tentam se apropriar do conhecimento intelectual coletivo para benefício próprio.

GNU/Linux



Sobre o GNU/Linux

O Linux é um *kernel* (núcleo) de sistema operacional criado pelo finlandês **Linus Torvalds** e mantido, atualmente, por vários desenvolvedores pelo mundo. O nome “Linux” veio de uma mistura do primeiro nome do criador: “Linus” + “Unix”. O Unix é um sistema operacional proprietário desenvolvido pela Bell Labs no início dos anos 70, e que serviu de base para uma variedade de outros sistemas operacionais.

O desenvolvimento do Linux teve início em 1991, quando Linux, na época estudante de Ciência da Computação na Universidade de Helsinki, Finlândia, resolveu desenvolver um sistema baseado no **Minix**, porém mais robusto. O Minix é um sistema operacional simples criado por Andrew S. Tanenbaum, com o objetivo de ser usado para estudos acadêmicos.

Então, por que chamar de **GNU/Linux** ? Acontece que Linus Torvalds desenvolveu a parte central do sistema operacional, conhecida como *kernel* , e o nome Linux deve ser atribuída somente a esse núcleo. Como ninguém vai utilizar um sistema operacional que só possui o *kernel* , era necessário para Linus Torvalds desenvolver os aplicativos a serem executados.

Enquanto Linus estava desenvolvendo o Linux, **Richard Stallman** objetivava a criação de um sistema computacional totalmente livre e, ao contrário de Linus, começou a desenvolvê-lo pelos aplicativos, como compiladores, editores de textos etc.

Foi da união dos aplicativos desenvolvidos pelo projeto GNU (de Richard Stallman) com o kernel do Linux (criado por Linus Torvalds) que surgiu o GNU/Linux, comumente chamado de Linux pela grande maioria das pessoas e também pelos meios de comunicação.

O bom desempenho, aliado à segurança e à possibilidade de personalização do sistema, devido ao código fonte ser aberto, tornou o GNU/Linux ideal para ser utilizado em servidores. Seu uso em máquinas de usuários comuns tem crescido bastante com a evolução das interfaces gráficas, o que está levando o GNU/Linux a ser o alicerce de muitos projetos de inclusão digital.



Distribuições GNU/Linux

Distribuições GNU/Linux são “estruturas” definidas através de uma metodologia que criam uma “personalidade” para este sistema operacional. São maneiras diferentes de agrupar o software necessário para se usar o sistema.

Algumas distribuições focam o uso do sistema operacional para um fim específico (servidores, super computadores, uso doméstico etc.). Outras procuram ser o mais abrangente possível, oferecendo tudo o que pode ser necessário (e até o que provavelmente nunca será) em sua distribuição. Algumas focam a estabilidade, outras o moderno, outras a quantidade de opções; algumas querem ser a mais amigável ao usuário, com a intenção de tornar a instalação ou o uso mais fácil, entre tantas outras características.

A distribuição ideal é aquela que melhor satisfaz as necessidades específicas do usuário. Cabe a cada um estabelecer quais são as suas necessidades e a partir daí buscar a distribuição que melhor atenda a seus desejos.

□ Debian

O projeto Debian foi iniciado em 1993 com um pequeno grupo de desenvolvedores e hoje conta com mais de 15.000 pacotes para mais de uma dezena de arquiteturas de hardware. O nome Debian (pronuncia-se “débien”)

é resultado da junção do nome de seu criador, Ian Murdock, com o de sua esposa, Debra.

Essa distribuição é conhecida pela sua grande estabilidade, segurança e por uma base sólida de gerenciamento de pacotes, conseguida por uma política de desenvolvimento rigorosa. O Debian possui sua própria definição de Software Livre, de forma que a licença de todo programa contido nele deve ser compatível com essa definição. Assim é garantido que 100% dos programas contidos no Debian são realmente livres.

O Debian é desenvolvido voluntariamente. Há aproximadamente mil desenvolvedores espalhados pelo mundo. Poucos se conhecem pessoalmente, sendo que quase tudo é decidido através de listas de discussão, e-mail e reuniões em salas de bate-papo IRC. Qualquer pessoa pode se tornar um desenvolvedor Debian, desde que contribua com o projeto de alguma maneira, tenha seu trabalho reconhecido por outros desenvolvedores e esteja de acordo com o Contrato Social Debian.

No Brasil existe uma distribuição personalizada Debian, o Debian-BR-CDD. Esta distribuição é atualmente baseada na versão estável do Debian (também chamada **Sarge**) e é voltada para usuários de Desktops domésticos que falam português. Ela traz, além de todas as novidades dessa versão, uma coletânea de pacotes especialmente feita para os usuários brasileiros, um instalador simplificado e um ambiente desktop amigável.

□ Ubuntu

Ubuntu é uma antiga palavra africana que significa “humanidade para todos”. O propósito da criação da distribuição é trazer esse espírito para o mundo do software. A construção é baseada nos princípios do Manifesto Ubuntu: o software deve ser gratuito, utilizável por qualquer um em sua própria língua e dando a liberdade de alterar e adequar às próprias necessidades.

O Ubuntu tem um sistema de gerenciamento de pacotes baseado no do Debian e pode utilizar seus pacotes, mas tem uma vantagem considerada importante pela maioria das pessoas: Um esquema de lançamento de versões em períodos regulares (de 6 meses), enquanto o Debian não lança uma versão enquanto ela não for considerada “pronta”. Atualmente oferece mais de mil pacotes de software e suporte a três arquiteturas.

□ Kurumin

Kurumin é um LiveCD (distribuição que roda direto do CD, sem precisar ser instalado no HD) nacional, baseado no Knoppix, que por sua vez é baseado no Debian. Bastante amigável (pois usa o gerenciador de janelas KDE), é ótima como “Meu primeiro GNU/Linux”. Possui uma gama de aplicativos para o uso doméstico como gravador de CDs, álbum de fotos digital, suporte a câmeras fotográficas digitais e web-cams, suporte a diversas impressoras, fácil conexão com a internet, aplicativos de escritório, jogos, etc. Essas qualidades, porém, têm um preço. Alguns softwares contidos no Kurumin não são livres e ele pode

apresentar grande instabilidade se instalado no HD.

❑ **Conectiva**

Conectiva Linux é uma distribuição nacional mantida pela Conectiva, baseada originalmente no Red Hat, mas que já se diferenciou o suficiente para ter sua “personalidade” própria. Essa distribuição foca o mercado latino-americano e para isso se esforçou para ter suporte para a maioria do hardware normalmente comercializado por aqui. Recentemente, a Conectiva fundiu-se a francesa Mandrakesoft, que também possui sua própria distribuição, a Mandrake, caracterizada pela facilidade de uso e pelo seu extenso conjunto de funcionalidades. A aglomeração das duas empresas ganhou o nome de Mandriva, mas até o momento ainda não lançou nenhuma versão da sua distribuição com o novo nome.

❑ **Slackware**

Slackware é uma criação de Patrick Volkerding surgida em 1993, o que faz dela uma das primeiras distribuições GNU/Linux. Visa a estabilidade e facilidade de uso, embora nem todos a considerem fácil de se usar. Um dos preceitos do Slackware é ser a distribuição mais parecida com o UNIX, tornando-a uma distribuição bastante personalizável, uma vez que quase não possui ferramentas automáticas de configuração como as outras distribuições. Embora a maior parte das pessoas ache que para instalar um programa no Slackware seja necessário compilar seu código fonte (embora isto seja sempre possível), uma vez

que não oferece um sistema de controle de pacotes, a distribuição vem com vários programas compilados e prontos para usar.

☐ Red Hat

É a distribuição comercial mais popular. Atualmente só dá suporte à versão Enterprise, voltada para servidores de grandes empresas. Porém, a Red Hat coordena um projeto chamado Fedora Core, que é mantido pela Comunidade de Software Livre, e é baseado na antiga distribuição da Red Hat para uso doméstico.



Interfaces Gráficas ou Gerenciadores de Janelas

☐ O que são “Interfaces Gráficas ou Gerenciadores de Janelas”?

Após a popularização do sistema operacional Microsoft Windows, o conceito de janelas parece ser óbvio num primeiro instante. Mas para aqueles que acompanharam o processo de amadurecimento das interfaces gráficas, a importância dessa tecnologia é mais evidente do que aparenta ser. Antes de continuar, consideremos que interfaces gráficas são o que visualizamos na nossa tela de computador, isto é, a aparência, representada por cores, formatos de botões, janelas e menus etc. Gerenciadores de janelas

são programas que controlam a interação entre esses diversos elementos, além de permitir a sua direta manipulação pelos usuários (como fechar, minimizar, arrastar janelas, abrir menus, executar programas etc). Os dois conceitos, portanto, possuem uma estreita relação entre si, já que enquanto um controla a aparência o outro controla a manipulação, e muitas vezes eles compõem uma peça só dentro do sistema.

No Windows, da Microsoft, o gerenciador de janelas não é uma peça separada do próprio sistema operacional, algo que ocorre quando vislumbramos o GNU/Linux. O Windows é um sistema no qual tudo, inclusive a interface gráfica, faz parte de uma estrutura inflexível, não estendível e pouco configurável.

No GNU/Linux, a interface gráfica se tornou uma realidade viável e acessível aos usuários leigos há pouco tempo. Porém, como o sistema é modular e a interface gráfica representa apenas uma parte estendida do sistema operacional, a sua manipulação e extensibilidade se evidencia na enorme quantidade de gerenciadores de janelas disponíveis na internet. Cabe ao usuário escolher aquela que mais lhe agrada e configurá-la a seu gosto.

Abaixo segue uma lista dos gerenciadores de janela mais populares:

GNOME



É o gerenciador de janelas oficial do Projeto GNU. Possui uma interface gráfica intuitiva e atrativa para

usuários finais e é altamente configurável. Desenvolvido por uma comunidade, é muito mais do que um gerenciador de janelas, já que possui centenas de aplicações associadas. Possui várias características que sinalizam uma preocupação crescente com a inclusão de usuários no mundo do Software Livre, como internacionalização e acessibilidade. Isso quer dizer que o GNOME é desenvolvido e documentado em diversos idiomas, além de possuir programas para classes especiais de usuários e pessoas com deficiências. O GNOME, porém é relativamente pesado, por isso não é recomendável para máquinas com hardware pouco potente.

KDE



O K Desktop Environment surge ao lado do GNOME como uma das opções mais populares, e os dois projetos são bastante parecidos. O KDE, porém, só ganhou mais força dentro da comunidade Software Livre depois que uma das suas peças chaves, a biblioteca Qt, passou de proprietária para livre. Tem subprojetos de desenvolvimento de software educativo, software para acessibilidade, entre muitos outros. Traz uma suíte de aplicativos de escritório própria, além de um bom gerenciador de arquivos, que também é usado como navegador para internet (*browser*).

Atualmente, o KDE vem se destacando por ser a interface gráfica nativa da distribuição Kurumin. Também não é recomendável para máquinas pouco potentes.

XFce



XFce é um gerenciador de janelas bastante leve desenvolvido para qualquer sistema operacional baseado no Unix, incluindo o GNU/Linux. Ele objetiva ser rápido e leve, enquanto mantém uma aparência agradável com facilidade de uso. O XFce consiste em um número de componentes que juntos provêem todas as funcionalidades do gerenciador de janelas. Esses componentes são empacotados separadamente, e o usuário pode escolher e instalar alguns dos pacotes disponíveis para criar uma interface gráfica mais personalizada. Utiliza as mesmas bibliotecas gráficas do GNOME.

XPde



O XPde é um projeto ainda novo que tenta aproximar os usuários do Windows XP para o mundo GNU/Linux. Para tanto a sua interface gráfica é parecidíssima com o Windows XP, apesar de manter as funcionalidades básicas de um desktop GNU/Linux.

WindowMaker



Este gerenciador de janelas foi criado por um brasileiro e tem sua interface gráfica toda baseada em quadriculos. Suporta integração com o Gnome e com o KDE, é rápido, rico em funcionalidades, super configurável e relativamente fácil de utilizar. Sua aparência

não se assemelha em nada com a da maioria dos gerenciadores de janelas — baseados em ícones e barras de tarefas. Traz o conceito de programas que rodam em mini-janelas que também são usadas como lançadores de aplicações. Possui um ótimo editor de configurações próprio chamado WPrefs, que facilita a configuração do ambiente, mas não tem um gerenciador de arquivos próprio. É um ambiente leve, que consome poucos recursos da máquina.

IceWM



O IceWM é um gerenciador de janelas prático e simples. Tão leve quanto o Window-Maker ou o Fluxbox, o Ice, ao contrário desses, tem uma interface muito mais próxima do que é considerado “convencional”: uma barra de tarefas na base da tela, com acesso ao menu de aplicações. Não usa bibliotecas gráficas pesadas, e não tem muitos recursos dos gerenciadores mais elaborados, como um gerenciador de arquivos, por exemplo. Ou seja, não é possível manipular arquivos de forma gráfica. Pela simplicidade de uso e pela leveza, tem sido a “segunda opção” (como alternativa ao gerenciador de janelas considerado pesado) em muitos projetos importantes.

Fluxbox



O Fluxbox é um gerenciador de janelas baseado no Blackbox e possui total compatibilidade com este. O Fluxbox herda, portanto, as características do Blackbox que por sua vez é um dos gerenciadores mais

rápidos e leves. Além disso, a sua instalação requer poucas bibliotecas, ao contrário da maioria dos gerenciadores de janela.



Tabela de Programas Equivalentes

Esta tabela é um resumo baseado na tabela de equivalências entre softwares para Windows e GNU/Linux coordenada pelo polonês Valery V. Kachurov. Alguns softwares listados na coluna GNU/Linux são proprietários, mas estão identificados como tais.

Tipo de Software	Windows	GNU/Linux
Escritório		
Editoração Eletrônica	Adobe PageMaker	Scribus, OpenOffice.org
Suite Office	MS Office, StarOffice	OpenOffice.org, GnomeOffice, KOffice
Processador de Textos	MS Word	Abiword, OpenOffice.org Writer, Kword
Planilhas	MS Excel	Gnumeric, OpenOffice.org Calc, Kspread
Apresentações	MS PowerPoint	OpenOffice.org Impress, Kpresenter
Banco de Dados	MS Access, Oracle, MS SQL Server	PostgreSQL, MySQL, Firebird, Pre-vayler
Finanças	MS Money	GNU Cash
Gerenciador de Projetos	MS Project	Planner (Mr. Project), Dotproject
Multimídia		
Tocador de Mp3/Ogg, estações de rádios Web	Winamp, Windows Media Player, MusicMatch Jukebox	XMMS, Mplayer, Xine, Rhythmbox
Gravador de CD	Nero, Easy CD Creator	K3b, XCDRoast, Gnome Toaster, WebCDWriter, CRecord
Tocador de CD	CD Player, Winamp, Windows Media Player	Grip, Gnome CD, Rhythmbox
Visualizador de Vídeos	Windows Media Player, RealPlayer, QuickTime, Winamp3	Mplayer, Xine, GXine, KDE Media Player, VLC
Tocador de DVD	WinDVD, MicroDVD, Windows Media Player	Mplayer, Xine, Aviplayer, Ogle, VLC, GXine, Totem
Extrator de mp3	MusicMatch, Real Jukebox	Grip, Lame, NotLame
Editor de áudio	SoundForge, CoolEdit	Ardour, Audacity, WaveForge, GNU-Sound, Glame, Sweep
Editor de vídeo (edição não-linear)	Adobe Premiere, Avid	Cinelerra, Kinox, Jahshaka
Notação Musical	Finale, Sibelius	LilyPond, Rosegarden
Gráficos		

Visualizador de imagens	ACDSee, Image Viewer	Eye of Gnome, GQView, Kview, Gthumb, GTKsee, Kuickshow
Visualizador de fotos (máquina digital)	Polaroid Drivers	Gtkam, Gphoto2
Editor de imagem (básico)	Paint	KPaint, Tuxpaint, Gpaint
Editor de Fotos	Adobe Photoshop, Corel PhotoPaint, Macromedia Fireworks	Gimp, ImageMagick, CinePaint
Editor de imagens vetoriais	Corel Draw	Inkscape, Sodipodi, OpenOffice.org Draw, Dia, Xfig
Modelagem 3D/Animação	3d Studio Max	Wings3d, Blender

Internet

Navegadores	Internet Explorer, Netscape	Mozilla, Firefox, Netscape, Opera, Konqueror, Epiphany, Sylpheed
Cliente e-mail	Outlook, Outlook Express, Eudora, Netscape Messenger	Evolution, Mozilla Messenger, Thunderbird, Kmail
Downloader de arquivos	Getright, DAP	Prozilla, Kget, Gnome Transfer Manager, GetLeft
Cliente FTP	CuteFTP	Gftp, Konqueror
Cliente IRC	Mirc	Xchat, Kirc, BitchX
Cliente de Mensagens	ICQ, ICQ Lite, MSN, AIM, Yahoo	Licq, Micq, GnomeICU, Gaim, Kopete, aMSN, Yahoo Messenger for Unix, AIM, Everybuddy, centericq
Video Conferência e VoIP	NetMeeting, Skype	GnomeMeeting, NeVoT, IVS, Skype (prop.)
Compartilhador de arquivos	Morpheus, Napster, Kazaa, eDonkey, Bittorrent	Lopster, Gnapster, eDonkey, Xmule, FreeNet, Bittorrent, Nicotine, LimeWire, aMule
Dial-up	Vdialer	Kppp, gtkdial, Gppp, Modem Lights

Manipulação de arquivos

Gerenciador de arquivos	Windows Explorer	Konqueror, Nautilus, gmc, Rox
Compactadores e descompactadores	WinZip, WinRar, arj, rar	Ark, zip, FileRoller, gzip, bzip2

Ferramentas Desktop

Visualizador de PDF	Adobe Acrobat Reader	GhostView, Xpdf, GV, Kghostview
Gerador de PDF	Adobe Acrobat Distiller	OpenOffice.org, PDFLatex, GV, GhostView, Xfig, Ghostscript, Kghostview
Programas de Scanner	Cds que acompanham o scanner	Xsane, Kooka
Anti-vírus	Norton Antivírus, TrendMicro, Dr. Web	OpenAntiVirus, Clam Antivírus

Jogos

Tetris	Tetris	Ltris, XWelltris
Minas	Mines	KMines, Perlmines, Dmines
Jogos de tiro	Doom (1, 2 e 3), Quake (1, 2 e 3), QuakeForge, DarkPlaces, Return to The Castle of Wolfenstein	Enemy Territory, Unreal 2003 e 2004, Quake (1, 2 e 3), QuakeForge, DarkPlaces (proprietários), Cube
Estratégia	Civilization, Sim City 3000, Warcraft 2	FreeCiv, Sim City 3000 (prop.), Stratagus
Corrida	Need for Speed	Tux Racer, KartlingRace
Simulador de Vôo	MS Flight Simulator	Simulador FlightGear
Lemmings	Lemmings	Pingus
Xadrez	ChessMaster	GChess, Xboard, Eboard

Inclusão Digital



O que é Inclusão Digital?

“Inclusão Digital” é a denominação dada, genericamente, aos esforços desenvolvidos por governos, organizações do terceiro setor e empresas no sentido de possibilitar às pessoas:

- ☞ obter os conhecimentos necessários para utilizar, com um mínimo de proficiência, os recursos das Tecnologias de Informação e da Comunicação existentes;
- ☞ dispor de acesso físico regular a esses recursos.

As estruturas e o funcionamento da sociedade contemporânea estão sendo significativamente alteradas pelas TIC de tal forma que favorecer a todos cidadãos o acesso a esses meios é um importante passo no combate à exclusão social.

No entanto, há uma discussão emergindo sobre o uso do termo inclusão digital. Existe, por exemplo, a crítica à banalização do termo, especialmente por conta da exploração política oportunista. Há também o “modismo” vinculado ao uso sensacionalista da expressão.

Mas já que se caminha por essas trilhas, seria enriquecedor apurar os significados atribuídos ao termo, para que assim se possa anular possíveis “explorações”, trazendo-o para um contexto sócio-político concreto do país, que o requisita em toda a sua profundidade e possibilidades. Aqui, estamos nos propondo a iniciar um percurso com esse objetivo.



Divagações sobre Inclusão Digital

Quer parecer politicamente correto? Fale em “inclusão digital”. Não porque a expressão se tornou recentemente banal de tão pronunciada, instituindo-se como um aparente modismo, mas pela relevância por ela mostrada. Em tempos da sociedade determinada pelo conhecimento adquirido e sua possibilidade de aplicação, o acesso à informação representa uma peça-chave, a partir da qual muitos caminhos dentro do ambiente social podem ser contemplados. À primeira vista, isso decorre do contato com as tecnologias da informação e da comunicação, seja através do reconhecimento inicial do conteúdo em formato digital, do uso de ferramentas tecnológicas existentes na produção de conhecimento novo, ou da construção de novas ferramentas capazes de atender às demandas permanentes

do mundo da informação.

Entretanto, um numeroso contingente de indivíduos está de fora deste já complexo universo sintético de consumo e produção de dados. Estes são os denominados “excluídos digitais”. Do mesmo modo que se convencionou tratar os desprovidos dos bens sociais básicos como marginais à sociedade estabelecida, os excluídos digitais se apresentam como os marginais aos meios de acesso à informação e geração de conhecimento. Com efeito, estes excluídos estão alheios ou dificilmente se revelam aos processos de alcance das necessidades subsistenciais, de cidadania, de interação social e de consciência do mundo em volta.

A inclusão digital possui o papel de resgatar os excluídos digitais ao contexto da sociedade movida pelos processos de criação, produção e sublimação da informação em conhecimento. Significa efetivar os excluídos digitais na sociedade da informação, por meio de políticas que visem ao seu crescimento auto-sustentável de forma colaborativa e gradual, não com medidas emergenciais e paliativas. Conseqüentemente, inclusão digital remete à busca da reflexão do mundo e da localidade, das condições de sobrevivência (emprego, alimentação, moradia etc.), do estímulo ao conhecimento renovado e à crítica do já existente, e da diminuição das desigualdades sociais.



Tabuleiros Digitais

Projeto desenvolvido na FACED/UFBA com o Liceu de Artes e Ofícios. Trata-se de um modelo de móvel que representa mais do que uma simples bancada para suporte de computadores - inspirado em soluções do cotidiano das trabalhadoras do acarajé.

O Tabuleiro Digital é reto, sem encostos, sem almofadas, projetado para uso rápido e ágil como o do tempo de comer um bom acarajé ou ler e responder meia dúzia de e-mails.

☐ Por que um tabuleiro digital?

- ☞ É simples de montar e pode não ter grandes sofisticções de acabamento. Foi desenvolvido para poder ser construído a partir de uma folha padrão de compensando, utilizando-se da sabedoria da baiana de acarajé. Basta cortar, lixar e pintar com verniz naval fosco as peças. Após a montagem, os tabuleiros podem ser dispostos em “ilhas” (agrupamento de tabuleiros isolados). É um modelo prático para adoção em espaços de acesso à internet, como infocentros, telecentros, ciberparques.
- ☞ Inclusão digital é mais do que ter acesso às máquinas. É o exercício da cidadania na interação com o mundo da informação e da comunicação. O Tabuleiro Digital, tal como os tabuleiros de acarajé da cidade de Sal-

vador da Bahia de Todos os Santos, espalha-se para proveito da comunidade universitária.



Acessibilidade

Quando falamos em acessibilidade, a idéia é tentar dar a **todos** a oportunidade de usar os recursos que o computador oferece. Porém, nem sempre esses recursos estão disponíveis para quem tem algum tipo de deficiência. Imagine que você tenha uma deficiência motora, que lhe impeça de mover um dos braços. Isso já pode constituir uma enorme dificuldade, pois além de tornar a digitação extremamente lenta, é impossível usar mouse e teclado ao mesmo tempo. Se os movimentos são restritos em ambos os braços, então a situação praticamente impossibilita o uso do computador.

Outra restrição grave são as deficiências visuais. Uma pessoa cega ou com dificuldades de enxergar, ou mesmo dificuldades de leitura, ou problemas para diferenciar as cores praticamente não tem recursos que lhe permitam usar o computador (com configurações convencionais) de forma eficiente. Algumas soluções são propostas para esses problemas.

A maioria das soluções proprietárias, entretanto, são muito caras, e algumas não são satisfatórias. Para deficientes motores, existem softwares de teclado virtual, que permite escrever utilizando o mouse, e softwares controlados por voz, que permitem acessar aplicações e menus

através de comandos de voz. Para deficientes visuais, há os leitores de tela.

Existem ainda opções de acessibilidade como criação de atalhos de teclado, teclas de alternância, teclas de repetição, de repercussão, bips de alerta etc. Essas opções podem ajudar bastante quem tem dificuldade de digitar ou possui restrições motoras que estão disponíveis tanto para os sistemas operacionais proprietários quanto para os livres. As dificuldades encontradas, entretanto, ainda são muitas. Páginas web mal feitas são praticamente impossíveis de serem lidas por alguns programas. Uma solução para isso é programar páginas para internet utilizando as técnicas de *tableless*, que deixam o código limpo e fácil de entender (tanto para os seres humanos quanto para o software que vai tentar interpretá-lo).

Para implementar um software que ofereça acessibilidade aos deficientes, é necessário que os programas com os quais ele tem que interagir (como editores de texto, menus etc) possibilitem uma certa integração com este software. Programas de código fechado normalmente não permitem que essa integração exista. O motivo é óbvio: se não temos como alterar um programa, e ele não foi projetado para permitir que fosse controlado por outra aplicação, nada é possível fazer.

Soluções livres existem, e muitas delas são bastante eficientes. As opções de acessibilidade do gerenciador de janelas KDE e do Gnome são ótimos exemplos.

A filosofia do software livre permite que o código dos programas seja alterado, de maneira que eles possam ser integrados com os programas que visam oferecer acessi-

bilidade. Assim eles podem, por exemplo, ser acessados e até controlados por outros programas. Isso facilita que sejam implementados teclados virtuais, controladores por voz, dispositivos de mouse alternativos (como aqueles controlados pelo movimento do globo ocular ou até por movimentos respiratórios).

Um fantástico exemplo de aplicação para deficientes motores é o *Dasher*. Ele permite escrever, mas não é exatamente um teclado virtual. Utiliza uma idéia inovadora para inserir texto através do teclado. É especialmente útil para escrever em *palmtops*, pois permite escrever muito mais rápido do que na forma tradicional (clitando sobre cada letra numa figura de teclado).

Na **seção de Acessibilidade do projeto Debian** (em www.debian.org/devel/debian-accessibility/index.pt.html) é possível encontrar muitas outras opções livres de acessibilidade, para as mais diversas necessidades.

Links



Mais sobre Software Livre

- ☞ Projeto GNU
www.gnu.org/home.pt.html
- ☞ Manifesto de Hipatia (Português)
www.hipatia.info/mh.pt.html
- ☞ Manual Foca Linux
focalinux.cipsga.org.br
- ☞ Noticias sobre o mundo Linux
www.noticiaslinux.com.br
- ☞ Noticias sobre o mundo Linux
br-linux.org
- ☞ Tabela de softwares equivalentes entre Windows e Linux
www.linuxshop.ru/linuxbegin/win-lin-soft-en

- ☞ Projeto Software Livre Brasil
www.softwarelivre.org
- ☞ Quilombo Digital
www.quilombodigital.org
- ☞ Portal de Software Livre do Governo Federal
www.softwarelivre.gov.br
- ☞ Telecentros livres
telecentros.softwarelivre.org
- ☞ PC Livre
www.pclivre.org.br
- ☞ Projeto Currículo Livre
curriculolivres.enec.org.br



Projeto Software Livre nos estados brasileiros

- ☞ Projeto Software Livre Bahia
www.psl-ba.softwarelivre.org
- ☞ Projeto Software Livre Ceará
www.psl-ce.softwarelivre.org
- ☞ Projeto Software Livre Distrito Federal
www.psl-df.softwarelivre.org
- ☞ Projeto Software Livre Espírito Santo
www.psl-es.softwarelivre.org

- ☞ Projeto Software Livre Goiás
www.psl-go.softwarelivre.org
- ☞ Projeto Software Livre Mato Grosso do Sul
www.psl-ms.softwarelivre.org
- ☞ Projeto Software Livre Minas Gerais
www.psl-mg.softwarelivre.org
- ☞ Projeto Software Livre Paraná
www.psl-pr.softwarelivre.org
- ☞ Projeto Software Livre Pernambuco
www.psl-pe.softwarelivre.org
- ☞ Projeto Software Livre Rio de Janeiro
www.pslrj.org.br
- ☞ Projeto Software Livre Rio Grande do Sul
psl-rs.softwarelivre.org
- ☞ Projeto Software Livre Santa Catarina
www.softwarelivre.sc.gov.br
- ☞ Projeto Software Livre São Paulo
www.psl-sp.org
- ☞ Software Livre no INPA - Amazonas
softwarelivre.inpa.gov.br
- ☞ Comunidade SOL - Software Livre - Amazonas
comunidadesol.org
- ☞ Movimento Software Livre Paraná
www.softwarelivreparana.org.br



Distribuições GNU/Linux

- ☞ Debian GNU/Linux
www.debian.org
- ☞ Debian-BR-CDD
cdd.debian-br.org
- ☞ Slackware Linux
www.slackware.org
- ☞ Suse Linux
www.suse.com
- ☞ Mandrake GNU/Linux
www.mandrakelinux.com
- ☞ Red Hat
www.redhat.com
- ☞ Fedora Core
www.fedora.redhat.com
- ☞ Knoppix GNU/Linux
www.knoppix.net
- ☞ Gentoo Linux
www.gentoo.org
- ☞ Conectiva Linux
www.conectiva.com.br
- ☞ Kurumin Linux
www.guiadohardware.net/kurumin/

- ☞ Outras distribuições (em inglês)
lwn.net/Distributions/
- ☞ Lista de distribuições que rodam direto do CD - LiveCD (em inglês)
www.frozentech.com/content/livecd.php



Interfaces Gráficas (Gerenciadores de Janelas)

- ☞ Gnome
www.gnome.org
- ☞ KDE
www.kde.org
- ☞ Xfce
www.xfce.org
- ☞ WindowMaker
www.windowmaker.org
- ☞ Fluxbox
www.fluxbox.org
- ☞ IceWM
www.icewm.org
- ☞ Outros gerenciadores de janelas (em inglês)
xwinman.org/others.php/



Alguns Softwares Livres bastante utilizados (conheça e use!)

☐ Escritório

- ☞ OpenOffice.org (em português)
www.openoffice.org.br
- ☞ Firefox e Mozilla (em português)
www.mozilla.org.br
- ☞ Abiword
www.abisource.com
- ☞ Gnumeric (do projeto GNOME)
www.gnome.org/projects/gnumeric
- ☞ KOffice (conjunto office do KDE)
www.koffice.org
- ☞ Scribus
www.scribus.org.uk

☐ Áudio e Vídeo

- ☞ Xmms
www.xmms.org
- ☞ Xine
xinehq.de

- ☞ Mplayer
www.mplayerhq.hu
- ☞ Totem (do projeto GNOME)
www.gnome.org/projects/totem

☐ Editores de Imagem

- ☞ Sodipodi (vetorial)
www.sodipodi.com
- ☞ Inkscape (vetorial)
inkscape.sf.net
- ☞ The Gimp (*bitmap*, com recursos de edição vetorial)
www.gimp.org
- ☞ Tux Paint (fantástico editor de imagens infantil)
www.newbreedsoftware.com/tuxpaint

☐ Jogos

- ☞ Armagetron
armagetron.sf.net
- ☞ Stratagus game engine
stratagus.sourceforge.net
- ☞ Chromium
www.reptilelabour.com/software/chromium/index.htm
- ☞ Portal de jogos GNU/Linux
tuxgames.gratishost.com

Realização

Visando o mesmo objetivo comum, de disseminação do software livre através de material impresso, duas organizações foram as responsáveis pela disponibilização deste documento:

- ☞ Projeto Software Livre Bahia – PSL-BA, autor original da cartilha
- ☞ Executiva Nacional dos Estudantes de Computação – ENEC, responsável pela distribuição



Sobre o Projeto Software Livre Bahia

O Projeto Software Livre Bahia (PSL-BA) é um movimento aberto a todos que busca, através da força cooperativa, disseminar na esfera estadual os ideais de liberdade difundidos pela Fundação Software Livre (FSF), possibilitando assim a democratização do acesso a informação,

através dos recursos oferecidos pelo Software Livre. Esta busca tem seus alicerces fundados na colaboração de todos, formando um movimento sinérgico que converge na efetivação dos ideais de Liberdade, Igualdade, Cooperação e Fraternidade.

Contribuindo com a Ciência, Tecnologia e Cultura – bases fundamentais para o desenvolvimento social – o PSL-BA empreende ações de discussão, difusão e efetivação da liberdade da informação e do conhecimento, almejando a conscientização e constante evolução de toda a sociedade.

O Projeto Software Livre Bahia é formado pela articulação de indivíduos que atuam em instituições públicas e privadas, empresas, governos ou ONGs, e demais setores da sociedade. Além disso o projeto não é subordinado a qualquer entidade ou grupo social, e não estabelece nenhuma hierarquia formal na sua estrutura interna.

O PSL-BA não coordena, não representa e não decide nada em relação à comunidade software livre da Bahia. A comunidade é um movimento totalmente amorfo, e assim o PSL-BA não arroja este tipo de liderança regional. O PSL-BA terá respaldo com a comunidade na medida em que seus participantes fizerem contribuições significativas para essa comunidade.

<http://www.psl-ba.softwarelivre.org>



Sobre a ENEC

A ENEC – Executiva Nacional dos Estudantes de Computação, fundada em 10 de Setembro 1993, é o órgão de associação, coordenação, orientação e representação dos estudantes dos cursos universitários de graduação e pós-graduação na área de computação do Brasil.

A ENEC é uma entidade civil, apartidária, sem fins lucrativos, gozando de autonomia administrativa, financeira e disciplinar, com prazo indeterminado de duração.

Segundo o MEC, a área de computação no ensino superior compreende os seguintes cursos: Ciência da Computação, Engenharia de Computação, Sistemas de Informação e Licenciatura em Computação.

No XI ENECOMP – Encontro Nacional dos Estudante de Computação, realizado na cidade de Florianópolis - SC, de 7 a 10 de Setembro de 1993, foi criada a ENEC. Nesta mesma ocasião foi aprovado o primeiro estatuto, definindo os objetivos da entidade, a forma de organização, a composição e atribuições da diretoria, forma de eleição, etc. Também foram definidas as atribuições da Assembléia Geral dos Estudantes de Computação, órgão máximo da ENEC.

□ ENECOMP

O ENECOMP – Encontro Nacional dos Estudante de Computação, é um evento anual, organizado pela direto-

ria da ENEC, pelas entidades de base locais do evento e por estudantes colaboradores. O evento é composto por mini-cursos, apresentações de projetos, palestras, mesas redondas e grupos de discussões sobre temas acadêmicos e sociais, como qualidade e currículos dos cursos de computação, regulamentação da profissão de informática, liberdade do conhecimento e software livre, além de temas ligados a novas tecnologias computacionais.

Durante o ENECOMP também é realizada a Assembléia Geral dos Estudantes de Computação. E ainda, com o objetivo de integrar os participantes, são organizados competições esportivas e festas.

□ XXIII ENECOMP

- ☞ Data: 01 a 05 de Agosto de 2005
- ☞ Local: Bonito – MS
- ☞ Informações: <http://www.enec.org.br/enecomp2005>

□ Assembléia Geral dos Estudantes de Computação

A Assembléia Geral é o órgão máximo da ENEC e sua sessão ordinária é realizada anualmente durante o ENECOMP, tendo como principais atribuições:

- ☞ Eleger e dar posse à Diretoria e ao Conselho Fiscal
- ☞ Alterar o Estatuto

- ☞ Deliberar soberanamente sobre os assuntos que surgirem e que sejam do interesse dos estudantes de computação.

Todos os estudantes de computação tem direito a voz na Assembléia Geral, mas apenas os delegados eleitos dos curso de computação tem direito a voto.

☐ Lista eletrônica de discussão

A ENEC, através da lista de discussão “enec-l”, tem atingido cada vez um número maior de estudantes de computação de todo o Brasil.

O endereço eletrônico da lista é:
enec-l@listas.enec.org.br

Para se inscrever, visite o site:
<http://listas.enec.org.br>

☐ Encontros Estaduais dos Estudantes de Computação

Os encontros estaduais são eventos que buscam organizar as entidades de base e os estudantes de computação nos estados brasileiros. O intuito principal desses encontros, além da integração e organização dos estudantes a nível estadual, é de também criar e amadurecer idéias, propostas e projetos para serem apresentados no ENECOMP.

O evento trata-se de uma ferramenta imprescindível para o fortalecimento da base da ENEC que são as Entida-

des de Base (CA's e DA's).

Os encontros estaduais adotarão a sigla:
ERECOMP-"sigla do estado"

☐ Executivas Estaduais dos Estudantes de Computação

Os estudantes de computação de cada estado brasileiro deverão compor a sua própria Executiva Estadual, com total autonomia e tendo seu próprio estatuto, onde ficará definido a composição da diretoria, suas atribuições, etc.

A criação das Executivas Estaduais facilitará o trabalho dos estudantes nos estados que discutirão seus problemas locais e ajudarão nos projetos nacionais da ENEC.

A Executivas Estaduais adotarão a sigla:
ENEC-"sigla do estado"

☐ Diretoria

Cada estado brasileiro possui um representante na diretoria da ENEC com cargo, atribuições e obrigações específicas. A forma de eleição é baseada nas entidades de base, que são os alicerces da ENEC.

O mandato da diretoria é de 1 ano e a eleição ocorre durante o ENECOMP.

□ Projeto Currículo Livre

O Currículo Livre tem por objetivo relacionar software e documentação livre para uso de estudantes e professores de cursos superiores da área de computação, assim como relatos de experiências de implantação de software livre nesses cursos.

O projeto pretende incentivar o uso de software livre dentro das Universidades, além de conscientizar os alunos e professores quanto ao modelo de desenvolvimento colaborativo de software e documentação, que é um processo aberto e democrático, permitindo e incentivando a participação de todos.

Sabendo disso, nada mais coerente que Universidades, que são espaços onde o objetivo-fim é a disseminação de conhecimento e a criação de novos saberes, façam uso de softwares que seguem esse modelo de desenvolvimento, que prima pela liberdade acima de qualquer coisa.

Através das liberdades oferecidas pelo software livre (executar o programa, estudar e modificar o código-fonte, copiar e redistribuir o programa), os envolvidos no processo ensino-aprendizagem no ensino superior podem estabelecer não apenas uma relação de aceitação dos softwares que são utilizados em sala ou laboratório de aula, mas verdadeiramente interagir com estes, com seus desenvolvedores e com a comunidade.

O projeto Currículo Livre iniciou com o currículo dos cursos de computação por ser uma necessidade conhecida dos envolvidos com o projeto desde sua concepção. Na forma de uma contribuição efetiva da ENEC à sociedade,

foi lançada a idéia de materializar a possibilidade de se fazer um curso completo de computação com a utilização apenas de softwares livre, como já acontece no curso de Ciência da Computação da Universidade Federal do Paraná.

A metodologia de trabalho do projeto consiste em, para cada disciplina "macro" do currículo dos cursos de computação (conforme especificado nas Diretrizes Curriculares dos cursos de computação do MEC), identificar pelo menos uma alternativa livre que possibilite tanto as atividades de ensino quanto futuras atividades profissionais naquela área da computação.

Assim, é disponibilizada uma compilação desse trabalho, de forma que instituições de ensino e professores interessados tenham uma referência para, em seus cursos de computação ou em disciplinas isoladas, utilizar softwares livres como ferramentas nas atividades de ensino.

Para que tal projeto seja bem-sucedido, é necessário o envolvimento desses mesmos estudantes e professores. Estes têm a consciência dos tipos de software que são utilizados hoje, e onde estão as principais carências ou mesmo desconhecimento de software livre já existente.

<http://www.enec.org.br>

enec@enec.org.br

<http://curriculolivre.enec.org.br>



Próximos eventos de software livre

II Conferência Latino-Americana e do Caribe de Desenvolvimento e Uso de Software Livre - LacFREE 2005

Local: Recife e Olinda - PE

Data: 28 de Setembro a 1 de Outubro de 2005

<http://LacFREE.softwarelivre.org>

III Encontro de Software Livre do Amazonas - ESLAM

Local: Manaus - AM

Data: 13 a 15 de outubro de 2005

<http://encontro.comunidadesol.org>

III Congresso Internacional de Software Livre - CONISLI

Local: São Paulo - SP

Data: 03 a 05 de Novembro de 2005

<http://www.conisli.org.br>

Latinoware Mercosul

Local: Curitiba - PR

Data: 19 e 20 de Novembro de 2005

<http://www.softwarelivreparana.org.br>

III Congresso Catarinense de Software Livre

Local: Florianópolis - SC

Data: 01 a 02 de Dezembro de 2005

<http://www.softwarelivresc.org.br>

Patrocínio