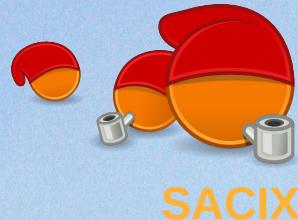


REVISTA

# espírito livre

LIBERDADE E  
INFORMAÇÃO

<http://revista.espiritolivre.org> | #002 | Maio 2009



SACIX

Terminais leves com Linux



TCOS PROJECT

Montando Thin Clients no  
Linux



DÚVIDA CRUEL...

Qual a melhor  
distribuição Linux?

# LEVEZA, ESTABILIDADE E PERFORMANCE

As distribuições Linux dão um  
show quando o assunto é isso!

## ENTREVISTA

Robert Shingledecker  
fala sobre o Tiny Core,  
seu novo projeto

## ISCA ANZOL REDE

Cuidado, muito cuidado  
com "certos" pescadores...

## DOTPROJECT

Gerenciamento de  
projetos com Software  
Livre



## Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 3.0 Unported

Você pode:



copiar, distribuir, exibir e executar a obra

Sob as seguintes condições:



**Atribuição.** Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.



**Uso Não-Comercial.** Você não pode utilizar esta obra com finalidades comerciais.



**Vedada a Criação de Obras Derivadas.** Você não pode alterar, transformar ou criar outra obra com base nesta.

- Para cada novo uso ou distribuição, você deve deixar claro para outros os termos da licença desta obra.
- Qualquer uma destas condições podem ser renunciadas, desde que Você obtenha permissão do autor.
- Nothing in this license impairs or restricts the author's moral rights.

Termo de exoneração de responsabilidade

Qualquer direito de uso legítimo (ou "fair use") concedido por lei, ou qualquer outro direito protegido pela legislação local, não são em hipótese alguma afetados pelo disposto acima.

Este é um sumário para leigos da Licença Jurídica (na íntegra).

# Chegamos para ficar!

Aqui estamos nós, depois do turbilhão e euforia quanto ao lançamento da edição número um, retornamos com uma edição ainda mais recheada de muita informação para você, leitor. Até o presente momento a primeira edição teve 8173 downloads, uma marca que a meu ver, para uma publicação com foco específico, merece menção. E tudo isso graças a você, que está a nos ler neste momento e que possivelmente teve de posse da edição de número um. Se ainda não leu, corra atrás da sua!

Nesta edição, você encontrará um assunto que rende muito pano pra manga: distribuições leves e com propósitos específicos, trazendo uma novidade: 2 entrevistas, a internacional, com Robert Shingledecker, criador do Damn Small Linux e mais recentemente o Tiny Core, ambas distribuições bem focadas no objetivo de ter uma distribuição leve, funcional e de alta performance em computadores com baixos requisitos de hardware. A entrevista nacional é com Flávio de Oliveira, responsável pela distribuição GoblinX Linux, uma distro bacana e descolada, que traz um visual bem moderno e igualmente leve. Outras duas matérias acompanham o assunto principal: Um review sobre o Sacix, a distribuição Linux utilizada nos telecentros do Projeto Casa Brasil e uma matéria super interessante sobre o TCOS - Thin Client Operating System, um projeto de porte e que merece toda nossa atenção.

Contamos novamente com a participação de nosso brilhante time de colunistas e colaboradores, porém a partir desta edição com novas adições. Isto mesmo! Assim que a edição de número um caiu na rede, pipocaram de pessoas de todo o Brasil (e de fora também) querendo colaborar, participar da equipe, ajudar na parte gráfica e escrevendo matérias. Enfim, foram tantos os pedidos que esperamos poder corresponder a todos da melhor forma possível. Já nesta edição, os leitores poderão conferir novas seções, novos rostos e novas perspectivas, trazidas por estes e tantos outros que estão aguardando seus materiais serem publicados.

Estamos a partir desta edição publicando uma seção extra de emails, onde publicaremos alguns dos emails mais legais que chegaram a nossa redação. Quer ver seu email ali publicado, então envie sugestões, opine, mande seu recado!

Continuem ligados no site da revista [<http://revista.espiritolivre.org>], onde além de encontrar a atual e as edições anteriores, sempre vão rolar novidades. Aproveitamos para avisar que buscamos parceiros em divulgação, entre outros. Sendo assim, se seu site estiver divulgando a revista através do nosso banner, avise-nos para que o incluamos em nossa seção de páginas amigas.

E continuamos aqui com o nosso contínuo pedido de materiais e colaboradores: você que quiser participar de alguma forma da Revista Espírito Livre, não se acanhe, entre em contato e faça parte deste time que está trabalhando duro pra montar uma publicação de qualidade para você, nosso leitor.

Chegamos para ficar e contamos com você!

**João Fernando Costa Júnior**  
**Editor**



## EXPEDIENTE

**Diretor Geral**

João Fernando Costa Júnior

**Editor**

João Fernando Costa Júnior

**Revisão**

Marcelo Tonieto

**Arte e Diagramação**

João Fernando Costa Júnior

**Capa**

Nilton Pessanha

**Contribuiram nesta edição**

Aécio Pires

Alexandre Oliva

Anderson Goulart

Andressa Martins

Cárlisson Galdino

Célio Maioli

Cezar Taurion

David Ferreira

Edgard Costa

Evaldo Junior

Filipe Saraiva

Flávio de Oliveira

Guilherme Chaves

Jomar Silva

Lázaro Reinã

Luiz Paulo

Marcello Duarte

Orlando Lopes

Paulino Michelazzo

Robert Shingledecker

Roberto Salomon

Sinara Duarte

**Contato**

[revista@espiritolivre.org](mailto:revista@espiritolivre.org)

O conteúdo assinado e as imagens que o integram, são de inteira responsabilidade de seus respectivos autores, não representando necessariamente a opinião da Revista Espírito Livre e de seus responsáveis. Todos os direitos sobre as imagens são reservados a seus respectivos proprietários.

# SUMÁRIO

## CAPA

### **33 Entrevista Nacional**

Flávio de Oliveira mostra a todos o GoblinX Linux e suas vantagens

### **37 TCOS Project**

Montando thin clients em Linux

### **41 Sacix**

Os desenvolvedores do SACIX mostram o projeto que visa criar terminais leves

### **44 Leveza+Versatilidade= Portabilidade+usabilidade**

Uma fórmula e tanto...

## COLUMNAS

### **11 Isca Anzol Rede**

Cuidado com "certos" pescadores

### **13 Fica no Trem!**

Mantenham todos a calma

### **15 Prontuário eletrônico de pacientes**

É o ODF ajudando a salvar vidas

### **17 Cordel do GNU/Linux**

E o Linux vira poesia...

### **20 Distribuições: Qual a melhor?**

Eis aí uma questão complicada...

### **22 Open source e a Academia**

Como o modelo open source poderia ser adotado na academia?

**PÁG. 25**



## **Entrevista com Robert Shingledecker**

Robert fala sobre sua vida, superações, Damn Small Linux e sobre o Tiny Core, seu novo projeto



**87 AGENDA**



**06 NOTÍCIAS**

## DESENVOLVIMENTO

---

### 46 Virado pra Lua - Parte 2

Que tal se tornar um "lunático"?

### 49 Aerotarget

Desenvolvendo games com software livre

### 54 Scrum

Uma aproximação ágil ao desenvolvimento de software

## ADMINISTRAÇÃO

---

### 57 Controlador de versões

Existem vários. Escolha já o seu!

### 59 Ocomon

Gestão de Incidentes com software livre

### 62 dotProject

Primeiras impressões sobre o software



09 LEITOR

## TECNOLOGIA

---

### 65 ODF - OpenDocument Format

Uma apresentação formal para os que ainda não o conhecem

## FÓRUM

---

### 70 Panóptico na Rede

A vigilância está crescendo e se fortalecendo...

## EDUCAÇÃO

---

### 74 Bicho-papão

A Matemática e o Software Livre

## EVENTOS

---

### 79 FLISOL 2009 no ES

Fique sabendo o que aconteceu FLISOL 2009 em Vitória/ES

### 82 FLISOL 2009 em Goiás

Sucesso de público em diversas palestras marcaram o FLISOL 2009 em Goiânia/GO

### 84 DIA LIVRE no Ceará

Idéias simples e ações práticas

# NOTÍCIAS

Por João Fernando Costa Júnior

## Lançado NetBSD 5.0



O responsáveis pelo projeto NetBSD lançaram a versão 5.0 desse sistema. O NetBSD é um sistema operacional altamente portável e é conhecido ainda por ser a segunda implementação open-source do BSD, depois do 386BSD. Esta versão vem com grande número de melhorias em performance e escalabilidade em multiprocessadores modernos e sistemas de múltiplos núcleos. Houve ainda outras melhorias como o alocador de memória "jemalloc", Xorg ao invés de XFree86 em alguns ports, o framework de gerenciamento de energia, ACPI suspend/resume, e muito mais. Para fazer o download desta versão a equipe prefere que seja feito a partir de [bittorrent](#), porém existem ainda os [espelhos](#).

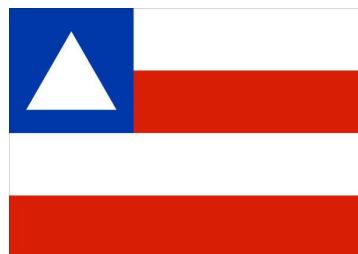
## Open3G: Discador 3G para Linux



Lançado o primeiro release público do discador para redes 3G da Open-Br para o GNU/Linux: o Open3G. Possui regras gerais para reconhecer os mais variados tipos de modems e tem as configurações de conexão das seguintes operadoras:

Claro, Vivo, Tim, Oi e Giro. É um beta público, com suporte nativo ao modem Sony Ericsson MD300 e ao Huawei E220, também testado nas operadoras Claro e Vivo. Ficou curioso? Então visite o [site oficial](#).

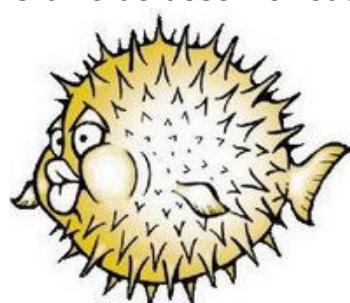
## Abertas inscrições para Encontro de Conhecimentos Livres da Bahia



Estão abertas as inscrições para o Encontro de Conhecimentos Livres da Bahia, que acontece de 11 a 14 de maio, no Junta Dados, Pontão de Cultura Digital da Bahia, situado no Pólo Universitário Santo Amaro de Ipitanga (Pusai/UNEB), em Lauro de Freitas. Durante o evento, a equipe do Centro de Desenvolvimento em Tecnologias Livres (CDTL), Pontão de Cultura Digital de Pernambuco, ministra oficinas de Áudio, Vídeo, Introdução ao Software e Cultura Livres e Metareciclagem, além de um workshop sobre Geração de Renda e Sustentabilidade. Clique [aqui](#) para preencher o Formulário de Inscrição ou acesse <http://www.tecnologiaslivres.org> para outras informações.

## Lançado OpenBSD 4.5

O time de desenvolvedores do OpenBSD lançou a versão 4.5 do sistema. Entre as novidades, estão um melhor suporte para a arquitetura Sparc64 e suporte a uma nova arquitetura, a ARM/OpenMoko, ainda várias melhorias de suporte à hardware e diversas funcionalidades novas, além de melhorias e limpeza de código fonte. O download pode ser feito na página de [download](#).



### Lançados oficialmente os GUFs - Grupos de Usuários Fedora



O Projeto Fedora Brasil informa a todos os interessados em contribuir com o Fedora que já estão criados para todos os estados Brasileiros os GUFs - Grupos de Usuários Fedora. Os GUFs serão organizados e coordenados pelos embaixadores e representantes do Fedora no Brasil. Esta iniciativa tem como objetivo a representação do Projeto Fedora em todos os estados brasileiros através da participação em eventos, distribuição de mídias, recrutamento de novos colaboradores e colaboração com os sub-projetos do Projeto Fedora Brasil. Confira os links das listas de discussão e comunidades em [http://www.projetofedora.org/guf\\_grupos\\_usuarios\\_fedora](http://www.projetofedora.org/guf_grupos_usuarios_fedora).

### InProprietário: o mundo do software livre



E o software livre vai às telas! Este documentário de cerca de 30 minutos sobre o Software Livre é o resultado do trabalho de conclusão do Curso de Comunicação Social do Centro Universitário FIEO, produzido pelo Daniel Bianchi e Johnata Rodrigo. Contém várias entrevistas, de algumas personalidades bem conhecidas do mundo do SL, visando explicar o que é o Software Livre, sua história, no que ele se diferencia do software não-livre como é o software proprietário ou o software grátis, qual a sua filosofia e o porquê de ele já ser um sucesso. O arquivo para download tem aproximadamente 300MB e está sendo distribuído livremente. Assistam, copiem, divulguem e distribuam este pequeno filme, é o que os seus autores pedem. O download pode ser feito via [torrent](#).

### Lançado FreeBSD 7.2



O time de engenheiros do FreeBSD lançou a versão 7.2 deste sistema tão popular. Ele vem repleto de novidades, incluindo suporte a uso completamente transparente de superpages para memória de aplicação, suporte para múltiplos endereços IPv4 e IPv6 para jails, csup agora suporta CVSMode para baixar um repositório CVS completo, Gnome atualizado para versão 2.26 e KDE para versão 4.2.2, sparc64 agora suporta processadores UltraSparc-III, entre outros. Para o download, clique [aqui](#), por meio de torrent. Ufa, esta última semana foi a semana BSD. O DragonFlyBSD 2.2.1 também foi lançado!

### Concurso para imagem nova versão 0.47 do Inkscape



O time do Inkscape e a comunidade a preparar o lançamento do Inkscape 0.47 e como já é de costume, foi solicitado que artistas ajudem na nova tela "Sobre" para esta versão. Todos os artistas estão convidados.

Para saber das regras, visite este [link](#).

### Mark Shuttleworth diz que Ubuntu não pode ser um Windows

Segundo Mark, "o Windows e o Linux rodam ambos aplicações importantes. O universo do Software Livre precisa desenvolver regras próprias. É um universo diferente do software proprietário. Nós precisamos alcançar o sucesso na nossa própria plataforma e nas nossas próprias regras. Se o Linux rodar apenas aplicações Windows, nos não poderemos vencer."

## A Fundação GNOME pede ajuda



**GNOME™** A crise econômica mundial que anda assolando o mercado também atingiu o GNOME. Para os que desconhecem, a GNOME Foundation serve como base para promover o GNOME Desktop. É ela que faz a ponte entre os patrocinadores (empresas e pessoas) e os desenvolvedores. Durante alguns anos a fundação teve um saldo positivo nas doações, o que permitiu a expansão das atividades relacionadas ao GNOME em eventos mundo afora. Só que agora a coisa anda mal. Para ajudar, visite o site oficial do Gnome e na página Amigos do GNOME: <http://www.gnome.org/friends>, e escolha a forma que deseja contribuir. Outra forma de colaborar é comprar camisetas e outros "apetrechos" oficiais. Saiba mais sobre a Fundação em <http://foundation.gnome.org/>.

## Novos conceitos: Menu circular no Gnome

Eis que aparece uma nova e interessante proposta para o menu do Gnome, o Circular Application Menu. Este menu abre-se em círculos e dele saem outros submenus circulares, e complementa-se com o Gnome-Do. Realmente uma proposta inteligente e com um design bastante futurista. O novo menu pode ser facilmente instalado a partir do [Getdeb](#). Atenção que é necessário o Compiz ou similar para funcionar. Para maiores informações, visite o [site](#) do projeto. Será que a era das barrinhas (verticais e horizontais) acabou?! Só o tempo dirá...



## SourceForge.net: Participe do Community Choice Awards



O SourceForge.net, um dos maiores repositórios (senão o maior) de aplicações de código aberto no mundo, organiza a cada ano uma votação para que a comunidade eleja os melhores projetos em código aberto. O prêmio chama-se SourceForge.net Community Choice Awards e a indicação de projetos para concorrer ao prêmio começou no dia 6 de maio, indo até o dia 29 deste mês. Os finalistas serão conhecidos em junho, sendo que a lista de ganhadores deverá sair no final de julho no evento [OSCON](#). Participe você também! Visite o site [Sourceforge.net](#) e vote.

## Lançado BrOffice.org 3.1.0



Após o lançamento do OpenOffice.org 3.1 (internacional), chegou a vez da versão nacional. O BrOffice.org 3.1.0 já pode ser baixado nos [mirrors](#) do projeto. Para baixar sua cópia em versões para Windows, Linux, Mac OS ou Solaris acesse: <http://ftp.uni-camp.br/pub/broffice/stable/3.1.0/>. BrOffice.org é uma suíte de programas de escritório totalmente gratuita utilizada por milhões de brasileiros. Para mais informações, visite o site oficial do [BrOffice.org](#).

Quer comentar sobre algo desta edição? O que gostou? O que não gostou? Participe! Envie seu comentário e/ou notícia para [revista@espiritolivre.org](mailto:revista@espiritolivre.org).



# EMAILS, SUGESTÕES E COMENTÁRIOS

Maxime Perron Caissy - sxc.hu

Esta seção foi criada para dividirmos com você leitor o que andam falando da gente por aí... Foram tantos emails e comentários no site da revista que ficou até difícil selecionar o que apareceria e o que não apareceria aqui. Muitos comentários dando uma força para nós da revista, realmente algo que dá um sentido todo especial ao trabalho desempenhado pela equipe. Ah! Continuem enviando suas sugestões e comentários!

Gostaria de dar os parabéns pela iniciativa, é a 1a vez que leo uma revista sobre opensource de tão alta qualidade. Temas modernos e bem abordados, a diagramação também está ótima. Faço questão de prestigiar aquilo é bem feito. A Revista Espírito Livre já está no meu bookmarks.

**Benjamim Góis - Belo Horizonte/MG**

Nesta manhã de segunda-feira em Vitória, pós feriado, trânsito enlouquecido pela paralisação do transporte coletivo e chegada tumultuada ao trabalho... recebo um email sobre a publicação digital da revista. Pensei em ler num intervalo de tempo durante o trabalho, resolvi abrir apenas o sumário, pronto! .....fui contaminada, como parar de ler artigos criativos como do Alexandre Oliva? A abordagem atual da colega Sinara Duarte sobre o Linux nas escolas, etc..

Em nossa rede municipal de ensino, o Linux veio para ficar e a cada dia, em cada escola, com cada educador nos deparamos com a Síndrome do Peter Pan Digital e me sinto como a Sininho, tentando espalhar o pozinho por onde passo e muitas vezes, tendo que voltar várias vezes para lançar novamente e tentar "contaminar" estes novos usuários. Mas como diz João Fernando: Vamo que vamo! Parabéns pela iniciativa e pela qualidade dos artigos.

**Karla Capucho - Vitória/ES**

Gostaria de parabenizá-los pela iniciativa. O conteúdo da revista está muito bom. Meu comentário é sobre o projeto de lei contra o cibercrime. Na minha opinião tal proposta agride diretamente o ideal da colaboração. É uma alternativa tão utópica que chega a ser absurda, muito distante da nossa realidade. Nossos políticos deveriam estar mais preocupados, por exemplo, com o projeto de lei PL-7109/2006, que fala sobre a regulamentação das profissões em Tecnologia da Informação. Fica como sugestão para um próximo artigo.

**Douglas Lima - Volta Redonda/RJ**

Só gostaria de comentar que achei fantástica a revista! Parabéns a equipe.

**Helton Eduardo Ritter - Três de Maio/RS**

## **COLUNA DO LEITOR**

---

Gostei muito da iniciativa de criar a Revista Espírito Livre. É bom saber que a cada dia mais usuários e entusiastas de softwares livres se reunem e promovem uma grande iniciativa como esta, contribuindo muito com a abrangência do tema, que a todo momento ganha mais espaço no mundo inteiro.

**Marllus de Melo Lustosa - Teresina/PI**

Conheci a Revista Espírito Livre hoje e gostei muito do estilo e das matérias. É um trabalho realmente bem feito, certamente com emprego de muito empenho.

**André Nunes - Salvador/BA**

Quero parabenizá-los pela revista. As matérias estão super atuais e abriram a revista com chave de ouro. A diagramação deixou o texto bem claro e não cansativo, ponto pra vocês. Sucesso!

**Tarcísio Cavalcante - Curitiba/PR**

Gostaria de parabenizá-los pela revista, linda, bem elaborada, leitura altamente dinâmica, show!

**Andrea Talarico - Ribeirão Preto/SP**

Gostei muito da revista se voltar para o software livre. Sou professor de informática e leciono para cursos técnicos e educacionais. Sempre busco utilizar softwares livres e propagar esta filosofia.

**Jefferson Marinho - São Paulo/SP**

Uma revista com assuntos diversos referentes a software livre. Inovando com assuntos jamais explorados em outras revistas online. Excelente publicação que está chegando com todo gás.

**Alan Messias - Camaçari/BA**

Gostaria de parabenizar a todos pela publicação da Revista Espírito Livre. Precisamos cada vez mais de iniciativas como essa para o bem da comunidade e para crescimento do SL.

**Corintho Fernandes Neto - Manaus/AM**

Nossa!!! Muito boa a revista. Sou um estudante de informática e estou começando agora no mundo open source. As matérias sobre a LPI e as dicas pra quem quer desenvolver software livre não poderiam ter vindo em melhor hora. Parabéns pela revista, além de ter um ótimo conteúdo, me deu bastante estímulo.

**Régis F. Brilhante - Mauriti/CE**

A revista Espírito Livre em sua primeira edição é uma ótima fonte de conhecimento e divulgação, traz material que atende desde o público iniciante ao avançado e de forma livre. Espero que continuem fazendo esse excelente papel, e que a comunidade ajude a divulgar o conhecimento. Agradeço a todos os contribuintes pelo trabalho. Estou ansioso para ler a próxima edição.

**Franzvitor Fiorim - Serra/ES**

Ótima conquista para o mundo do software livre.

**Ricardo Martiniano - João Pessoa/PB**

Meus parabéns pela iniciativa, não tenho dúvidas que esta revista vai estourar em todo Brasil.

**Carlos Henrique - Brasília/DF**

Saudações! Gostaria de expressar minha total satisfação pelos temas que a revista "Espírito Livre" aborda. Realmente o conteúdo visa fazer com que o leitor compreenda o texto sem que este seja um profissional do ramo. Empolgante também é o fato dela ser livre, sendo livres podemos ir mais adiante e isto é muito estimulante. Que a força esteja com vocês, sucesso!

**Caiocesar Bayerl Fornaciari - Iconha/ES**

Já baixei e já li, sensacional. Parabéns pela iniciativa e desejo muita sorte na nova empreitada.

**Ariovaldo Carmona - São Paulo/SP**

# Isca Anzol Rede

Por Alexandre Oliva

Os usuários de software de maior consciência social já notaram que, ao morderem a isca, incentivam os pescadores de usuários a continuarem se valendo de anzóis escondidos para capturá-los. Não por acaso, pescadores têm recorrido à Grande Rede como mecanismo alternativo para aprisionamento. Que fazer para evitar esses perigos?

## Hora da bóia!

A história começa com um apetitoso pedaço de camarão, minhoca ou qualquer outra funcionalidade que possa atrair usuários. No mesmo pacote, cuidadosamente projetado para ludibriar a presa, vem o anzol, característica maliciosa destinada a restringir e aprisionar o usuário, de forma técnica (negação de código fonte, DRM, Tivoização) ou jurídica (contratos restritivos, licenças limitantes, ameaças

diversas).

Mesmo quem entende e valoriza a liberdade, e portanto faz o possível para evitar os anzóis, é às vezes vitimado por esses dispositivos feitos para enganar e aprisionar. A diferença é que, uma vez fisgados, alguns concluem que o melhor que lhes resta a fazer é saborear a isca, relaxar e gozar a viagem, enquanto os mais conscientes batalhamos para nos libertar até o fim das forças, mesmo sacrificando recursos importantes e convenientes, tais como partes da boca, do computador ou de sítios.

Muito mais inteligente é não mordermos a isca: além de não corrermos o risco de captura, ainda desanimamos o pescador. Quanto menos usuários cairmos na armadilha, maior a chance de os pescadores abandonarem essa estratégia, de modo que todos possamos

voltar a nos alimentar, com tranquilidade e segurança, de bits de camarão e de informação.

Pescadores do software, anti-éticos que são, já encontraram outra estratégia para ludibriar suas presas, inclusive se beneficiando do movimento solidário dos usuários, o Movimento Software Livre, contra a captura imediata por meio de anzóis.

## Lança-se a Grande Rede

Os espertinhos começaram a semear as iscas no éter digital, usando a Grande Rede para envolver os cardumes de usuários atraídos por elas, conduzindo-os, desprevenidos, até a captura definitiva.

Curioso e preocupante é que a Grande Rede pode aprisionar o usuário de duas maneiras: tanto no microambiente do próprio usuário

quanto pelo acesso ao sítio do pescador.

Na primeira, a isca não fica presa a um anzol, nem ao ambiente do usuário: o pescador envia-lhe as iscas goela abaixo diretamente de seu sítio. São iscas com tecnologias AJAX, Flash e Java, entre outras. O usuário as processa nas próprias entradas, sem muita chance de adequá-las às suas preferências e necessidades. Distraído pelas iscas, nem percebe que a rede vai ocupando cada vez mais espaço ao seu redor. Pior que é o pescador quem decide o que a vítima consumirá, a cada visita. Nessas condições, as iscas são um perigo mesmo que suas receitas sejam originalmente Livres. Pior ainda com as descaradamente não-Livres, que usam Ofuscript às centenas de toneladas.

Na segunda maneira, o pescador estende a rede entre usuários e um sítio convidativo para depositarem e manterem seus ovos e informações pessoais. Mesmo que não sejam capturados, usuários correm o risco de que, mais tarde, a passagem não esteja mais livre; de que suas crias nasçam em cativeiro.

Antes do advento da rede, o usuário podia defender pessoalmente suas crias de ameaças externas, mantendo-as sob seu controle em seu próprio micro-ambiente isolado. Já no sítio, é o pescador quem controla o ambiente e o aces-

so às crias, podendo utilizá-las para fins indesejados ou até oferecê-las a terceiros. Sem controle, o usuário fica à mercê do pescador, mesmo que possa obter receitas e especificações do ambiente (como requer a GNU Affero GPL) e configurar seu próprio ambiente adaptado em sítio alternativo.

Até no caso de uma aplicação colaborativa como, digamos, a fecundação dos ovos, é essencial para os usuários que eles, e não um pescador, possam controlar o acesso aos ovos e o ambiente em que são mantidos. Não bastam as especificações do ambiente do sítio mantido pelo pescador. Se os usuários não puderem levar suas crias para outro sítio, elas se tornarão reféns, usadas para controlá-los.

### Nada é perfeito?

Os pescadores têm muitíssimo interesse em controlar as gerações atuais, mas não lhes basta. Esperam imbuir nas gerações que nascem e crescem sob sua influência a aceitação desse desrespeito à liberdade. Inocentes usuários, por sua vez, parecem não perceber o risco para si mesmos e suas crias. E que faz um usuário desavisado para evitar os perigos das iscas, dos sítios e das redes dos pescadores? Nada! Fica por ali mesmo, crédulo e crente no conforto e na segurança do ambiente controlado pelo pescador.

É verdade que o futuro do usuário capturado pela Grande Rede ainda tem chance de não ser, digamos, nebuloso. Afinal, há pescadores que devolvem suas presas à liberdade, ainda que às vezes com sequelas. Mas há os que capturam prisioneiros para que vivam em cativeiro. Uma vez no aquário, pular fora é difícil e perigoso, possivelmente fatal, especialmente para as crias deixadas para trás. É melhor evitar! E que faz um usuário consciente para evitar os perigos das iscas, dos sítios e das redes dos pescadores? Nada! Nada para bem longe deles! 🐟

Copyright 2009 Alexandre Oliva

Cópia literal, distribuição e publicação da íntegra deste artigo são permitidas em qualquer meio, em todo o mundo, desde que sejam preservadas a nota de copyright, a URL oficial do documento e esta nota de permissão.

<http://www.fsfla.org/blogs/lxo/pub/isca-anzol-rede>



**ALEXANDRE OLIVA** é engenheiro de Computação e Mestre em Ciências da Computação. Usuário, desenvolvedor, evangelizador de Software Livre e contribuidor do projeto GNU desde a década de 90. Engenheiro de Compiladores na Red Hat desde fevereiro de 2000. Co-fundador e conselheiro da FSFLA - Fundação Software Livre América Latina.



# Fica no trem!\*

Por Paulino Michelazzo

Estava frio, um frio de lascar. Daqueles que dói nos ossos e as mãos ficam azuis. Tudo bem, era por uma boa causa afinal não é sempre que pode-se conhecer uma cidade de uma civilização secular perdida no meio do nada incrustada entre montanhas das mais altas e vegetação grandiosa.

Mas existia algo que suspirava a possibilidade de melhor a oportunidade única. A luta pela liberdade se formava diante de nós e corria o boato de acabar com o passeio de todos. De um lado, manifestantes reivindicando não sei o que e, de outro, o símbolo máximo da ordem social; a polícia com seus escudos, capacetes e

bombas de gás. Um palco armado com boa dose de adrenalina correndo nas veias por estarmos no meio do confronto.

Passam os minutos e nada daquele que ia nos levar para a cidade sagrada aparecer. Alguns, diziam, o trem estava parado na cidade aguardando o desfecho do evento. Outros acreditando que não iria aparecer trem nenhum e outros ainda pensando estar cometendo o maior erro de suas vidas. O sol começa a aparecer e aproveita-se para esquentar a pele e respirar um pouco de ar fresco fora do ônibus que mais parecia lotação paulistana em manhãs de chuva. Neste meio

tempo não somente o sol esquenta, mas também os ânimos de ambos os lados e começa, do nada, a chuva.

E a chuva era diferente, feita de pedras e bombas de gás. De um lado, nativos mandando ver bem ao estilo clássico de futebol e de outro, menin black com suas bazucas flamujantes. No meio, quem? Os maiores defensores da liberdade do software mundial enfurnados na já comentada lotação, esperando por aquele projétil que iria entrar pelo vidro.

No meio da baderna um “gênio” tem a iluminada idéia:

- Quero sair!

- Mas porque quer sair no meio desta loucura toda?

- Quero sair! Nunca senti cheiro de gás lacrimogênio na vida. Quero sair!

Sob protestos de todos, lá desce o mais estranho de todos para, numa fungada de dar inveja a qualquer baleia, respirar um pouco da fumaça que estava ao redor do ônibus e voltar em menos de um minuto com os olhos vermelhos sanguineos e a garganta queimando como o inferno. Dizem que quem avisa amigo é mas também cachorro velho não aprende truque novo...

Passada a pancadaria, chega o trem e todos embarcam rapidamente para quem sabe não se tornarem alvos ambulantes. Minutos depois, a velha máquina diesel começa

seu caminho e depois de hora e meia, faz uma pequena parada numa antiga estação para que subam alguns outros passageiros. Neste momento para a surpresa de todos (ou de quase todos), o mesmo estranho resolve descer do trem para uma “rápida esticada nas pernas”.

- Não desça, protestam alguns. O trem irá partir e não poderá voltar

- Mas eu quero descer...

- Não desça!

E quem disse que deu certo? Lá vai ele caminhando pelo vagão e desce à plataforma para seu tranquilo caminhar no meio do fim do mundo. Como verdadeiros videntes, o trem logo depois começa seu chocalhar sem nosso insólito e teimoso passageiro que fica perdido há milhares de quilômetros de casa sem falar o idioma local e somente do notas de dólar.

Seguimos viagem e duas horas após chega aquele que, já tão famoso pelos seus feitos no software livre, agora era famoso pela sua excentricidade em terras latinas. Avistando-o ao longe comento com meu amigo e compadre: veja lá quem saiu da tumba! E este, mais que depressa e com seu humor afiado, retruca: que sorte, seria o primeiro GNU perdido nestas bandas do planeta. No mínimo daria uma pesquisa da National Geographic dentro de alguns anos.

Curioso para saber como fez para voltar a nos acompanhar, descubro que o velho GNU foi resgatado da plataforma de trem por um grupo de nacionais que vinham na composição seguinte e viram aquela inconfundível figura a caminhar feito gato em dia de mudança se questionando: o que vou fazer da vida?

Pelo menos a história tem uma vantagem; em todo o trajeto de retorno não tivemos nenhuma surpresa. Parece que a síndrome do “esqueceram de mim” deixou-o cabisbaixo.

\* A história é verídica e passa-se em Cusco/Machu Picchu no Peru em 2003



**PAULINO MICHELAZZO** é diretor da Fábrica Livre ([www.fabricalivre.com.br](http://www.fabricalivre.com.br)), empresa especializada em soluções para Internet com ferramentas de gestão livres. Foi diretor mundial da Mambo Foundation, System Development Specialist da ONU no Timor Leste. É instrutor de CMS's (Drupal, Joomla e Magento). Escreve regularmente para diversos canais na Internet e publicações técnicas no Brasil e em Portugal.

# Padrão ODF e o Prontuário Eletrônico de Pacientes

Por Edgard Costa

Lemos e ouvimos no rádio e televisão, todos os dias, o desafio que é, para o cidadão brasileiro, conseguir obter uma vaga em uma unidade de saúde ou mesmo uma consulta médica.

Planos de Saúde, juntamente com as operadoras de telefonia móvel ou fixa, são campeões de reclamações no PROCON. O governo brasileiro vem tomando algumas iniciativas, ainda tímidas, para racionalizar e equacionar estes problemas, criando a Agência Nacional de Saúde (ANS) para fiscalizar as operadoras de Planos de Saúde que, em tempos remotos, eram monitoradas por uma diretoria do Banco Central especializada em seguros. Imagine, leitor, que Cooperativas Médicas ( exemplo Unimed ), quando criadas, eram registradas no Ministério da Agricultura, porque só este órgão público regulava a criação de empresas cooperativas. Por isto tem, ain-

da, que exibir em seus logotipos esta imagem padrão no cooperativismo. 

A Agência Nacional de Saúde, recentemente, lançou a primeira iniciativa governamental para implementação futura do Prontuário Eletrônico de Pacientes, o T.I.S (Troca de Informação na Saúde Suplementar), que tem como principal finalidade fazer as Operadoras de Planos de Saúde informem, em tempo real, os procedimentos médicos a que estão se submetendo seus associados.

Estudos conduzidos por Universidades Brasileiras mostram que, se adotado um Prontuário Eletrônico de Pacientes Universal, o governo brasileiro poderá economizar algumas centenas de milhares de reais em procedimentos desnecessários ou solicitados sem nenhum critério específico.

A adoção do PEP gerará um imenso banco de dados,

dando condições para o Ministério da Saúde realocar os recursos, tanto humanos como em espécie, para áreas em que, realmente, são necessárias, melhorando sua aplicação e fiscalização, o que certamente fará com que o atendimento ao cidadão brasileiro se torne, minimamente, civilizado.

A adoção de um Prontuário Eletrônico de Pacientes não se dará de forma pacífica. Entre os problemas que deverão ser enfrentados podemos destacar os seguintes:

- a) Regionalismos Lingüísticos;
- b) Facilidades na manipulação de arquivos de papel;
- c) Personalização da codificação do Prontuário;
- d) Pouco gasto em treinamento de pessoal auxiliar;
- e) Validade Jurídica, comprova da, dos arquivos de papel;
- f) Gastos imensos com implementação de plataformas de Hardware e Software;
- g) Gastos imensos com imple-

mentação de Redes de Computadores. Tanto locais como externas;

- h) Falta de cultura na utilização do computador, tanto pelo profissional de saúde bem como do pessoal auxiliar;
- i) Preço do certificado digital e hardware para a sua utilização;
- j) Segurança e privacidade na manipulação das informações médicas.

Os argumentos contrários aos da resistência são:

- a) Redução de custos;
- b) Redução na redundância de Exames;
- c) Reconstrução da história do Paciente de forma completa;
- d) Facilidade Organização;
- e) Racionalidade de Espaço;
- f) Interoperabilidade;
- g) Intercâmbio de layout de dados;
- h) Processamento contínuo dos dados;
- i) Integração com outros sistemas de informação;
- j) Dados atualizados.

E o ODF em que lugar entra nesta história complexa?

O ODF (Formato aberto de documentos) já é um padrão ISO oficializado e conhecido mundialmente como os demais que compõe o PEP que são:

- 1.DICOM – para imagens médicas;
- 2.HL7 (Health Level seven) para comunicação entre os módulos que compõe o sistema

hospitalar;  
3.PKCS-12 – padrão para certificação digital.

Não seria plausível imaginar que softwares para PEP usassem padrão de documentos proprietário por razões clássicas. Vou citar apenas uma. Imagine não se conseguir abrir um determinado laudo radiológico determinante para um ato cirúrgico porque não se tem a mais nova versão do software editor como aconteceu quando houve a tragédia no continente Asiático motivada por um terremoto, em que países não puderam enviar ajuda humanitária para a região atingida porque não conseguiram abrir, em seus computadores, os ofícios encaminhados por não terem a versão atualizada deixando centenas de pessoas com sede, fome e sem remédios.

Outra razão é que o padrão ODF já está pronto para usar certificados digitais.

Como todos sabem, o ODF é largamente programável, facilitando em muito a geração de grandes quantidades de relatórios, da automação do envio de documentos por email, de imagens, criação da versão final do PEP que deverá ser em PDF- A, e interoperável.

Além disto tudo, há a razão maior. A financeira. A rede pública tem que, obrigatoriamente, aplicar os escassos recursos na atividade fim e não comprando sofisticados siste-

mas para gerar e gerenciar o Prontuário Eletrônico de Pacientes. Ela poderá gerar este PEP utilizando uma ferramenta já muito conhecida, em várias instâncias do governo, que é o BrOffice.org que se integra, perfeitamente, com várias plataformas de banco de dados, gera formulários com extrema rapidez e com forma visual agradável para o usuário, pode ser instalado e atualizado por demanda, se integra muito bem com python, via Macros ou APIS para manipulação de imagens médicas.

Enfim, outros tempos para a saúde estão por vir. Claro que os desafios para implementação de um projeto auspicioso, que determine um tratamento digno para o cidadão brasileiro, são incomensuráveis. Como já diziam os sábios chineses: “ Para uma longa jornada temos que dar o primeiro passo”. Estamos começando a andar. E andar na direção certa.

### Maiores informações:

#### Site Prontuário Eletrônico:

[www.prontuarioeletronico.odo.br](http://www.prontuarioeletronico.odo.br)

#### Site ODF Alliance Brasil:

<http://br.odfalliance.org>



**EDGARD COSTA** é membro do Grupo de Usuários BrOffice.org do Estado de S.Paulo, Assurer Cacert – Certificação Digital e autor do Livro BrOffice da Teoria à Prática.

# Cordel do GNU/Linux

Por Carlisson Galdino



Lynne Lancaster - sxc.hu

Literatura de cordel é um tipo de poesia popular especialmente no Nordeste brasileiro. Tradição de Portugal, os livretos deste tipo de poesia eram vendidos em feiras, pendurados em barbante (ou cordel).

O Cordel do GNU/Linux é escrito em sextilhas em rima x-A-y-A-z-A, usando redondilhas menores (versos de cinco sílabas poéticas).

## Cordel do GNU/Linux

Tem gente que pensa  
Que computador  
É calculadora  
Mal lhe dá valor  
Mas ele é bem mais  
Que pode supor

Tem gente que pensa  
E pensa saber  
O que é o negócio  
Chamado PC  
Pensando que é só  
Máquina de escrever

Mas computador  
É bem mais que isso  
É um equipamento  
Robusto e preciso  
Que é diferente  
De tudo que é visto

Para funcionar  
Precisa programas  
E o programador  
Vivia um drama  
Sem poder dormir  
Tranquilo na cama

O computador  
E um bicho danado  
Dentro tanta coisa  
Fora outro bocado  
Monitor de vídeo  
E mouse e teclado

E aqueles lugares  
De botar disquete  
CD ou outra coisa  
É coisa pra peste!  
Memória e circuitos  
Bios, chipset

Cada fabricante  
Já que é seu direito  
Cada componente  
Fará do seu jeito  
Eis o pesadelo  
Que já estava feito

Pois antigamente  
Cada programinha  
Tinha que saber  
A história todinha  
E usar do PC  
Tudo o que ele tinha

Um programa feito  
Rodava somente  
Num computador  
Pra outro diferente  
Teria que ser  
Feito novamente

Como um instrumento  
Feito por medida  
Que não funcionava  
Em outra guardada  
E isso complicava  
De todos a vida

Foi quando alguém  
Teve uma sacada  
Fazer uma coisa  
No centro instalada  
Pra cada programa  
Não precisar nada

Essa coisa estranha  
No canto central  
Chamou-se Sistema  
Operacional  
Ou Operativo  
Lá em Portugal

É esse programa  
Que quebra a cabeça  
Pra saber usar  
Tudo o que apareça

No computador  
Cada placa ou peça  
  
E os outros programas  
Conversam com ele  
De um jeito padrão  
Sem muito enfeite  
E o Operacional  
Faz o papel dele  
  
Assim um programa  
Pra grande espanto  
Não era como antes  
Pois com grande encanto  
Feito só uma vez  
Roda em qualquer canto

Pra ter um Sistema  
Operacional  
Chamado S. O.  
Temos afinal

Umas opções  
Como é normal

O mais conhecido  
Se chama Windows  
Mas diversidade  
É um negócio lindo  
E não tem só ele  
E há outros surgindo

Inclusive um  
O melhor que tive  
Falaremos dele  
Que ouça quem vive!  
E o melhor de tudo  
É software livre

Sobre soft livre  
Falei outro dia  
Do nosso sistema  
Fala esta poesia  
Para quem não sabe  
Ou pouco sabia

Havia um sistema  
Que se utilizava  
Na Universidade  
E se apreciava  
Era S. O. Unix  
Como se chamava

Uma confusão  
Num tempo confuso  
Mudou o cenário  
E impediu seu uso  
Pelo copyright  
Ou foi seu abuso

Para resolver  
Tão triste questão  
Um novo projeto  
Surgiu logo então  
O Software Livre  
Teve uma Fundação

GNU Não é Unix  
Era este o projeto  
Criar um S. O.  
Unix aberto  
Era o objetivo,  
Perfeito e completo

Assim foi nascendo  
Foi bem natural  
Surgiu um Sistema  
Operacional  
Dando liberdade  
Ao usuário final

Free Software Foundation  
Ou FSF  
Criou o GNU  
Embora tivesse  
Faltado uma coisa  
Que ela fizesse

Ainda não disse  
Pra não confundir  
Se você entendeu  
Tudo até aqui

Vamos com cuidado  
Então prosseguir

Pois é que um Sistema  
Operacional  
Tem dentro de si  
Pra ser funcional  
Muitos programinhas  
E uma parte central

A parte central  
De todo S. O.  
É chamada kernel  
Não funciona só  
Mas é necessária  
Senão, tenha dó...

Claro que a FSF  
Disso bem sabia  
Então planejou  
Uma engenharia  
Bem sofisticada  
Pro que ela queria

Mas esse tal kernel  
Nunca ficou pronto  
E longe do States  
De um outro canto  
No país Finlândia  
Veio um novo espanto

Um kernel foi feito  
Por prazer, não dor  
Aberto e robusto  
Como se sonhou  
Chamado Linux  
Devido ao autor

O kernel Linux  
Cresceu bem ligeiro

Um dia encontrou  
Num golpe certeiro  
O S. O. GNU  
Se uniu por inteiro

Feitos um pro outro  
Corpo e coração  
GNU e Linux  
Fizeram união  
Assim se tornaram  
Úteis desde então

O tempo passou  
Do norte ao sul  
Ele é utilizado  
Sob o céu azul  
Mesmo que esqueçam  
O nome GNU

Pois chamam Linux  
O Sistema inteiro  
Esquecem GNU  
Que veio primeiro  
O Linux que é  
Dele um parceiro

Mas o importante  
É a qualidade  
Que o sistema traz  
E a liberdade  
E sem falar que  
Vírus não o invade

Há muita opção  
Pra quem quer usar  
GNU com Linux  
Pra então se livrar  
Do S. O. fechado  
Pra se libertar

O famoso Debian  
Que uso desde antes  
O Ubuntu, que é bom  
Para iniciantes  
Fedora, Mandriva  
E muitos restantes...

Espero que tenha  
Entendido o recado  
Sobre esse Sistema  
Que já é comentado  
Mas o assunto é um pouco  
Mesmo complicado

Adeus a quem leu  
Com isso se importe  
Se quiser tentar  
Esse S. O. forte  
Desejo a você  
Boas vindas, boa sorte!



**CARLISSON GAUDINO** é Bacharel em Ciência da Computação e pós-graduado em Produção de Software com ênfase em Software Livre. Já manteve projetos como IaraJS, Enciclopédia Omega e Losango. Hoje mantém pequenos projetos em seu blog Cyaneus. Membro da Academia Arapiraquense de Letras e Artes, é autor do Cordel do Software Livre e do Cordel do BrOffice.

Em Salvador - Bahia,  
Dois Grandes Eventos  
29 e 30 de Maio.  
[Entrada Grátis]



**III Encontro Nordestino de Software Livre**  
**IV Festival de Software Livre da Bahia**



# Distribuições: Qual a melhor?

Por Roberto Salomon

Svilen Mushkatov - sxc.hu

Não sei se acontece com todos mas inevitavelmente, em uma conversa com alguém no trabalho, surge a dúvida: qual a melhor distribuição? Responder o inevitável "depende" parece apenas reforçar a dúvida e acabo puxando uma cadeira pois acaba sendo uma conversa para dois barris de chope.

Cada um de nós tem uma distribuição do coração. Quando duas pessoas começam a discutir "distros", parece até discussão de time de futebol. Ninguém concorda. E, quando concordam em uma, não conseguem concordar no porquê.

Longe de mim pretender esclarecer este ponto mas não posso perder a chance de dar a minha opinião sobre o assunto. Como ex-usuário de SuSE (do jeito que se escrevia na época) e atual usuário de Ubuntu (pronto, acabei de esclarecer, logo de cara, qual a distribuição que uso) e já tendo trabalhado com Conectiva, Mandrake, Mandriva e Red Hat, acho que posso comentar um pouco sobre distribuições sem correr muitos riscos.

Vamos começar pelo que é, formalmente, uma distribuição. Uma distro é uma coleção

“ ... a melhor distribuição é aquela que atende a sua necessidade.”

Roberto Salomon

de software mantida por um grupo com interesses mais ou menos comuns. Digo mais ou menos porque não importa qual distribuição adotada, tem sempre aquele aplicativo que não está disponível. Apenas essas diferenças de conteúdo costumam suscitar discussões muito interessantes sobre qual a melhor distribuição e já ouvi argumentos de todos os tipos sobre o por que da distro “A” ser melhor que a distro “B” só porque tem este ou aquele pacote.

Quando o universo das distribuições começou a se consolidar, elas logo se dividiram em dois grandes grupos: as baseadas na Red Hat e as baseadas em Debian. E as discussões começaram para valer. (Antes que me joguem pedras, não me esqueci do Slack nem do Knoppix ou mesmo do Gentoo, todas distribuições extremamente competentes mas que, apesar de contar com usuários apaixonados, não ga-

nharam o volume que as “duas grandes” conseguiram). Os argumentos de pacotes existiam, mas logo o argumento da tecnologia usada para distribuir aplicativos ganhou o palco na comunidade mais técnica. Passamos a discutir qual o melhor: RPM ou DEB.

E o mundo continuou a andar. E as comunidades a discutir.

Hoje temos as distribuições especializadas nos mais diversos nichos. Tenho estudado algumas distribuições como o DSL (Damn Small Linux) que se propõe a montar um ambiente completo em menos de 50MB ou o Coyote que monta um kernel com pilha de rede e firewall para cargas de até 200 usuários em um ambiente mínimo. Isso sem falar da nova mania nacional: as distribuições “remixadas” para os netbooks.

Quem vê de fora não entende como conseguimos conviver com tantas distribuições. Uma para cada dia da sema-

na, segundo um conhecido meu. O que estas pessoas não entendem é que a existência de tantas distribuições diferentes apenas comprova a força da comunidade e do próprio conceito do Software Livre. Não apenas sou livre para escolher os softwares que quero usar como também sou livre para escolher como quero empacotá-los.

Acho que acabei divagando um pouco e deixei no ar a principal dúvida: qual a melhor distribuição? A resposta? Eu não sei. Para mim é a que estou usando no momento. Para uma empresa, deve ser aquela que oferece o suporte mais confiável. Para outros, aquela que empacota o maior número de jogos ou a que roda naquele modelo de telefone celular.

No final da história, a melhor distribuição é a que atende à sua necessidade. 

### **Maiores informações:**

**Blog do Roberto Salomon:**  
<http://rfsalomon.blogspot.com>



**ROBERTO SALOMON** é arquiteto de software na IBM e voluntário do projeto BrOffice.org.

# OPEN SOURCE E A ACADEMIA

Por Cézar Taurion



Outro dia fiquei imaginando como o modelo Open Source poderia ser adotado na academia. Na minha opinião, o modelo de ensino tradicional adotado hoje por muitas universidades ainda assume um processo de aprendizado formal, geralmente individualizado e centrado no professor (teacher-centric). Os alunos são mediados por testes e fazem, ao fim do ano o famoso TCC (trabalho de conclusão do curso) de forma individual. Colaboração e team-work não são valorizados nestes testes! São chamados de cola e combatidos... E na maioria das vezes os testes não colocam o problema no contexto, tendem a ser teóricos e fora da realidade do dia a dia do futuro profissional. Para mim este modelo deve ser re-

pensado!

Mas, não é só. Existe outro grande desafio no ensino de disciplinas de computação. A evolução tecnológica e a demanda de conhecimento evoluí muito rapidamente. Imaginem um curso de graduação de quatro anos em ciência da computação. Excetuando-se as disciplinas básicas, metade do que o aluno aprende nos primeiros anos estará obsoleto lá pelo terceiro ou quarto anos. Os mecanismos de atualização dos cursos atuais ainda estão, em sua maioria, adaptados aos tempos pré-Internet e não conseguem acompanhar na velocidade adequada a evolução tecnológica.

Olhemos agora as comunidades Open Source. Elas são

orgânicas ou sejam, auto organizadas, e incentivam processos informais de aprendizado em grupo. Podemos classificá-las como "learner-centric". Na minha opinião, os projetos Open Source são um belo exemplo de ecossistemas de aprendizado, pois as comunidades Open Source conseguem prover e distribuir, de forma sustentável, o conhecimento necessário para a produção de software de boa qualidade. Além disso, do ponto de vista do aprendizado, o compartilhamento de informações transcende os skills de programação, havendo extensa troca de idéias entre os membros das comunidades em assuntos diversos como patentes, licenças, habilidades gerenciais e principalmente tra-

balho em equipe.

Se analisarmos suas principais caraterísticas, veremos que muitas delas são muito desejáveis para um processo de ensino mais prático em computação.

Querem alguns exemplos?

a) Conteúdo gerado pelo usuário. Porque os estudantes não podem contribuir pró-ativamente para a criação e evolução do material do curso, através de wikis, código fonte, blogs, etc? O engajamento ativo dos estudantes aumenta sua motivação e abre perspectivas inovadoras para o conteúdo do curso.

b) Atividade real. Fazermos os alunos contribuirem com código real para uma comunidade Open Source existente ou a ser criada pelos próprios alunos, é um trabalho útil e uma experiência profissional sem preço. Eles passam a ter contato com outros profissionais e estudantes (do mundo inteiro) e aumentam sua percepção e prática do que é desenvolver software de forma colaborativa. O compartilhamento de informações com outros estudantes e profissionais é altamente benéfico. Aqui enfatizo o ponto que alunos de graduação tem condição técnica de colaborarem com código. Este código não necessariamente será aceito pela comunidade, mas será um esforço que valerá pena.

Participar ativamente de uma comunidade Open Source significa usar tecnologias intensamente, que variam de wikis, listas de discussão e chat, até ferramentas de geração e gerenciamento de código.

Cézar Taurion

c) Uso intenso de tecnologias. Participar ativamente de uma comunidade Open Source significa usar tecnologias intensamente, que variam de wikis, listas de discussão e chats, até ferramentas de geração e gerenciamento de código. Uso prático e real. Aliás, muitos alunos têm blogs e usam redes sociais. Seria usar estes mesmos mecanismos nas salas de aula.

Um aspecto importante que faço questão de enfatizar é o potencial de aprendizado prático que se tem quando se trabalha em projetos reais Open Source. Pelo que vejo e ouço, existe uma defasagem grande entre o aprendizado na maioria dos cursos de computação e a necessidade do mercado. Muito do que o aluno aprende nas escolas não é crítico ao dia a dia nas empresas, faltando a eles um maior experiência prática. Os exemplos prá-

ticos devem ser focados em casos reais e não puramente hipotéticos.

Mas como adotar o modelo Open Source no ensino da computação? Talvez o primeiro passo seja fazer com que algumas disciplinas passem a demandar contato direto com projetos Open Source. Estas disciplinas, e aí podemos falar em inúmeras atividades, como desenvolvimento de programas, aprendizado em banco de dados, sistemas operacionais, etc, exigirão que os alunos contribuam, de forma colaborativa, para comunidades Open Source. E nem precisam ser atividades de geração de código. Podem ser contribuições para os wikis das comunidades e mesmo até contribuições para o Wikipedia. Os mecanismos atuais de ensino estão defasados diante desta nova demanda. Muita gente ainda se assusta com o

fato de trabalhos em grupo permitirem a alguns se esconderem e não trabalharem. Mas as ferramentas de colaboração como wikis podem apontar quem colabora e quem não colabora.

Bem, as condições básicas para se implementar o modelo Open Source na academia já existem. Temos milhares de projetos Open Source que necessitam de maior colaboração para evoluírem. Identificar os projetos em que vale a pena investir tempo e dedicação, já é por si, um processo de aprendizado.

As tecnologias adotadas pelos projetos Open Source são, obviamente, Open Source, o que significa que não existe custo adicional de compra

de licenças para as universidades. O resultado para o aprendizado dos alunos será altamente positiva e a experiência que eles obterão nestas disciplinas será bastante útil para sua vida profissional. Portanto, o que falta para isto acontecer?

Difícil responder. Mas, a primeira barreira é a quebra de paradigmas. Conversei com alguns professores e vi alguma resistência. Uns acham que é utópico pensar em trabalhos colaborativos. Outros não consideram que os alunos de graduação estarão aptos a desenvolverem código. Mas, porque não experimentar? Seguir os caminhos que outros já trilharam só vai nos levar aonde os outros chegaram. Não além. 

### Maiores informações:

#### ODF Alliance:

<http://www.odfalliance.org>

#### Site do OpenOffice.org:

<http://www.openoffice.org>

#### Site do BrOffice.org:

<http://www.broffice.org>



**CEZAR TAURION** é Gerente de Novas Tecnologias da IBM Brasil. Seu blog está disponível em [www.ibm.com/developerworks/blogs/page/ctaurion](http://www.ibm.com/developerworks/blogs/page/ctaurion)

São Paulo • 21 de junho de 2009



**WordCamp**  
Brasil



# Mega-entrevista com Robert Shingledecker, criador do Tiny Core Linux

Por João Fernando Costa Júnior

Robert Shingledecker fala com a Revista Espírito Livre sobre seu novo projeto, o Tiny Core, uma distro de apenas 10 MB. Robert é conhecido por ter sido um dos principais desenvolvedores do famoso Damn Small Linux. Durante a entrevista, ele fala de seu novo projeto, vivências, fatos relevantes que ocorreram no passado, mas principalmente o que ainda está por vir.

**Revista Espírito Livre:  
Apresente-se aos nossos leitores!**

Robert Shingledecker: Me chamo Robert Shingledecker. Vocês devem conhecer o meu trabalho, porque eu fui o maior colaborador do Damn Small Linux. Fui também o co-autor da Prentice

Hall, Chris Negus Linux Series, "The Official Damn Small Linux Book". Meu mais novo projeto é o Tiny Core Linux. Estou com 60 anos, aposentado, e resido em Fullerton, Califórnia. Possuo mestrado em matemática pela Universidade Estadual da Califórnia. Sofro de uma deficiência de último estágio nos olhos (Distrofia Muscular Oculofaríngea). Quando posso, gosto de escrever e compartilhar códigos com os outros. Isso mantém minha mente afiada e longe de tornar-se deprimida com pensamentos sobre meus problemas de saúde. Entretanto, durante toda minha carreira, por 36 anos, eu sempre promovi o compartilhamento de conhecimento sobre computadores, linguagens e software.

## REL: Há quanto tempo trabalha com linux e software livre?

RS: Tanto meu trabalho quanto meu passatempo tem sido "brincar" com computadores e linguagens. Minha primeira experiência com computadores foi na Burroughs Corp. em 1971 como um contratado independente. Mexi com linguagem de código de máquina (sim, em hexadecimal) e programas de contabilidade em fita de papel perfurado. Quando optei por um trabalho formal de 9 às 17, fui para cidade de Garden Grove, Califórnia, onde apresentei a cidade ao Samba em abril de 1995, com Linux hospedando uma rede Windows 3.11. Esta foi a primeira implantação em larga escala de Linux nos Estados Unidos. Recebi visitantes de várias partes do globo

interessados em ver a distribuição. Eu me encontrei com muitos iluminados do free software, Linus, Stallman, John Maddog Hall, Bob Young, Gael Duval, e fui convidado a falar na primeira Linux Expo World, em San Jose, expondo a respeito de nossa distribuição criada para Linux. Tive muitos eventos como esse em seguida. Geralmente era para debater sobre redes da Novell. O maior evento em que discordei foi a COMDEX no outono de 1999. Sai da cidade em janeiro de 2000, para me tornar o CTO de uma pontocom. Realmente eu era o CTO de diversas pontocom. Foi nesta pontocom que eu e outro programador criamos appliances em Live CDROM com Linux, incluindo um desktop em LiveCD. Vocês podem visitar meu Web site,

<http://www.shingledecker.org>, para muito mais detalhes sobre todos meus projetos de Linux.

## REL: O Damn Small Linux (DSL) é incrivelmente popular. Como foi criado? De onde vieram as idéias por trás da criação do DSL?

RS: Minhas primeiras anotações a respeito do DSL são datadas de três de setembro de 2003. Eu tinha encontrado o DSL no DistroWatch e, como muitos eu suponho, fiz minha própria versão. Muitos podem não saber, mas o fundador e dono da marca registrada DSL, John Andrews, começou o DSL usando um disco de recuperação de 50 MB do KNOPPIX "KNX". Removeu as aplicações de recuperação e as trocou por pequenas aplicações GTK de desktop, algumas das quais eram do Deli Linux. Eu me cadastrei no forum do DSL em setembro 2003 e mandei por email para John com algumas correções e sugestões. Inicialmente estes eram para pendrives USB e clientes NFS. Muitos meses depois contribuí com uma opção de backup e restore, um bash\_profile funcional, o bootlocal.sh, e um diretório de /opt com suporte a escrita. Em janeiro de 2004, me deram a tarefa de criar e manter a imagem ISO do DSL. Em seguida, eu criei pessoalmente todos os releases desde a versão 0.5.3.1 de 15 de janeiro de

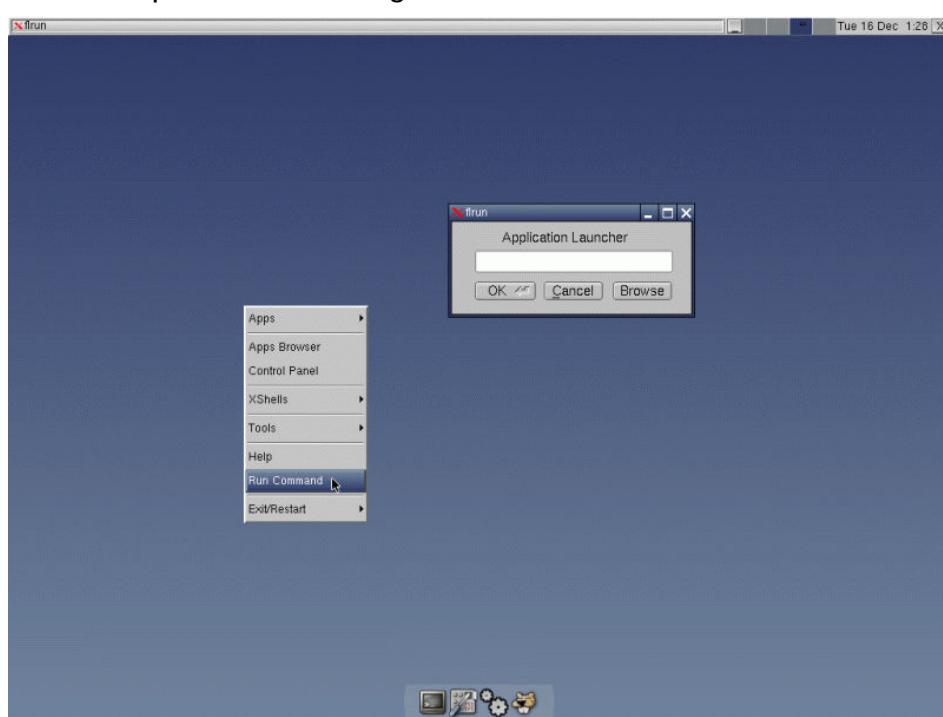


Figura 1 - Tiny Core Linux em execução

2004. Cada melhoria estrutural do DSL era minha criação. Eu estava tentando trazer características novas a um LiveCD de modo que houvesse pouca ou nenhuma vantagem em executar uma instalação tradicional no HD (isto foi bem antes do UnionFS). Escrevi um artigo intitulado "Not your father's Operating System" (em português, "Não é o Sistema Operacional do seu pai"), em que expliquei muito de minha filosofia. Continuei com a instalação frugal, usei o GNU coreutils para substituir o BusyBox, escrevi a documentação "Getting Started"

nei aos ícones de clique único ou duplo, um Icon Layout Manager e um desktop com suporte a arrastar e soltar. Adicionei suporte a Winmodem para os dispositivos populares da Lucent, suporte para adicionar usuários, implementei o UnionFS e o tipo de extensão UNC, configurei os repositórios, foi criada a versão PXE, criei disquetes de inicialização com suporte a USB e suporte a PCMCIA. Codifiquei e implementei vários novos códigos novos de boot como secure, protect, legacy, desktop, icons, e waitusb. Criei as bibliotecas usadas por bash scripts e Lua, assim como o pendrive install scripts, ZIP e HDD. Criei a possibilidade de se usar o dpkg-restore, suporte a multiusuário para instalação no HD, e web backup e restore. Eu desmontei também o KNOPPIX duas vezes (a v3.3 para DSL v0.6 e depois a v3.4 para o DSL v2.0) e compilei o kernel 2.4.31 e todos os módulos para manter o DSL atualizado. O

DSL que você está usando hoje não importa que seja 0.6+, 1.x, 2.x, 3.x ou 4.x, foi primeiramente o resultado de meus esforços com a exigência que eu devia tornar real no DSL as aplicações em GTK. Isto para que o DSL fosse sempre visto como o DSL. Em outras palavras, não era para eu mexer com os aplicativos.

**REL: E o Tiny Core?  
Fale-nos sobre este seu novo projeto. O que você esperava criar?**

RS: Quando eu sou perguntado "por que mais uma outra distribuição?", isso me faz recordar do Linux World 2005. Ele estava lá, quando eu tentei mostrar o DSL, eu fui desacreditado com uma observação do tipo "Oh sim, você é todo menino de treze anos tem sua própria distribuição Linux". Mas quando este fornecedor viu o que eu tinha, ele percebeu que não iria nadar no mar da mesmice. Minha filosofia é oferecer uma maneira original de rodar o Linux. Eu não promovo instalações tradicionais no HD. Eu chamo agora esse método de instalação de "scatter mode" (algo como modo esparramado, espalhado). Você não o encontrará listado no Core Concepts methods do Tiny Core. Com o tempo o desempenho é impactado ou se torna corrompido. Seja através do mau funcionamento do sistema de software ou hardware, erro do usuário/operador, o

 Minha filosofia é oferecer  
um maneira original de rodar o  
Linux. Eu não promovo instalações  
tradicionais no HD.  


Robert Shingledecker

(em português, Guia de Introdução), criei e apresentei a extensão de sistema MyDSL, assim como as extensões compactadas montáveis em loop, que mais tarde vieram a se chamar UCI. Introduzi o Lua/FLTK (Flua) para criar umas 50 GUI front ends (interfaces gráficas) para o DSL. Eu também integrei o QEMU ao DSL e criei uma versão separada do SYSLINUX. Adicio-

tecas usadas por bash scripts e Lua, assim como o pendrive install scripts, ZIP e HDD. Criei a possibilidade de se usar o dpkg-restore, suporte a multiusuário para instalação no HD, e web backup e restore. Eu desmontei também o KNOPPIX duas vezes (a v3.3 para DSL v0.6 e depois a v3.4 para o DSL v2.0) e compilei o kernel 2.4.31 e todos os módulos para manter o DSL atualizado. O

que quer que seja. Eu acredito que a inicialização de um computador deve ser rápida. Deve sempre partir de um conhecido estado imaculado. Acredito que se deve ter o controle dos processos que estão funcionando na hora da inicialização e na coleção das aplicações que se deseja usar. Visto que, a maioria das distribuições está se tornando cada vez maiores e maioria delas oferecendo mais atrativos para os olhos do que funcionalidades e ditando o ambiente de execução e a seleção de aplicativos. Acho estes processos muito lentos, desnecessários demais, e não trazem as aplicações que eu deseo. As outras distribuições pequenas ditam ainda uma coleção das aplicações. A maioria que eu nunca uso. O gênesis do Tiny Core foi a partir de uma reunião na Linux World 2005, comigo, Kent Porter, e Chris Livesay. Nós discutimos o que pensávamos ser um ambiente ideal. Um que facilmente sustentaria os conceitos e a filosofia que eu tinha introduzido no Damn Small Linux, mas sem o peso adicionado daquelas aplicações em GTK. Deixe o usuário decidir-se; GTK, ou GTK2, linha de comando para servidores, desktop light, ou appliance especializada. Eu criei realmente o Tiny Core quando eu desmontei o KNOPPIX 4 e fiz o DSL-N. John Andrews rejeitou o conceito. Ele, outra vez, ficou do lado das aplicações do DSL-N, de modo que o Tiny Co-

re não fosse liberado. John abandonou logo DSL-N e assim, a seguir, eu o fiz. Com não muito mais do que eu poderia fazer com o kernel antigo 2.4 do DSL, eu procurei mais uma vez encontrar uma base nova. Foi até que vi o SliTaz e me lembrei das conversas com Rob Landley no OLS2006 sobre o popular o Initramfs com o BusyBox. Eu o estudei, junto com os registros de desenvolvimento do kernel e vi como um disco inicial de RAM simples e um Busybox poderiam trabalhar com minhas idéias originais e conceitos de extensões. Eu carreguei o Finnix, uma poderosa e pequena distribuição baseada em Debian, e o kernel de Landley seguido do Rob e o Busybox para criar o primeiro protótipo. A seguir comecei a mergulhar no código que eu tinha criado durante meus últimos cinco anos do DSL, para fazer o primeiro Linux Tiny Core. A iteração seguinte era uma conversão do (refatorado) murgaLua para C++/FLTK para chegar a um protótipo funcional do Tiny Core.

O Tiny Core não é um fork do DSL. Ele tem uma base completamente diferente e não é nem baseada no Debian, nem no KNOPPIX. O Tiny Core não é também um remaster de SliTaz, mas foi feito baseado nas novas potencialidades do kernel 2.6 junto com as características que o Busybox forneceu. Embora seja pequeno (10MB), o Tiny Core não é focado em nenhuma parte específica do hardware. É injusto dizer que por causa do tamanho o Tiny Core é para um hardware mais antigo.

### REL: Como o Tiny Core trabalha?

RS: O Tiny Core é contido inteiramente em um arquivo comprimido cpio que popula o disco inicial de RAM em cima do booting do Kernel Linux. Então, basicamente o Tiny Core é composto por dois arquivos: bzImage (o Kernel do Linux) e tinycore.gz. Por causa disso o Tiny Core roda inteiramente na RAM, o que é muito rápido e

“ “ É injusto dizer que por causa do tamanho o Tiny Core é para um hardware mais antigo.” ”

Robert Shingledecker

permite também que nos ofereça diversas opções para persistência. Isto tudo está documentado online na seção Core Concepts do nosso Web site. A idéia é uma separação dos dados estáticos versus dados dinâmicos. Todos os dados estáticos, tipicamente aplicações, são empacotados sob um arquivo TAR (TCE) ou em imagens comprimidas para a montagem em loop (TCZ). Estes pacotes, nós os chamamos de extensões, e estão disponíveis em nossos repositórios online. Adicione agora a aqueles dois arquivos, o bzImage e tinycore.gz um diretório chamado "tce" e o Tiny Core vai mesclar ou montar um diretório inteiro das extensões durante o boot. O resultado é um desktop feito sob demanda baseado na sua escolha das aplicações. Os dados dinâmicos, tipicamente seus dados pessoais, situados em seu diretório home são persistidos com um backup e um restore. Isto também pode ser automático e é dirigido por dois arquivos .filetool.lst e .xfiletool.lst. Aqueles que estiverem

familiarizados com o comando TAR irão perceber que o arquivo .filetool.lst é para a opção "T" e arquivo .xfiletool.lst é para a opção "X". Estes dois arquivos são populados previamente para facilitar a utilização, mas podem facilmente ser editados para um controle mais "fino". Este é apenas um dos modos de operação para o Tiny Core. Nós oferecemos diversos outros, todos constituindo diferentes níveis de persistência. Um exemplo seria instalar as extensões em um arquivo ou em um diretório de loop, assim eliminando a carga em cima do tempo de boot. Ou ainda usando um home persistente para evitar o backup e o restore.

O Tiny Core sempre é carregado de uma imagem comprimida cpio. Assim, cada boot é como se fosse o primeiro boot do CDROM. Na verdade nós sugerimos que os arquivos do Tiny Core estejam colocadas em um HD, numa instalação frugal, pequena e organizada. O Tiny Core pode facilmente coexistir com uma distribuição

existente do Linux. Basta copiar o bzImage e o tinycore.gz em seu HD e ajustar o seu bootloader, neste caso, o GRUB. Adicione um diretório tce e você está pronto para rolá-lo. Mesmo usando o modo persistente ele usará um diretório /home já existente e adicionará simplesmente um diretório "tc" no home. Assim a cada novo boot o sistema estará em um conhecido estado imaculado. Nós não promovemos fazer uma instalação tradicional do HD. Eu chamo-o de "scatter mode" ou modo esparramado, disperso, porque não é pequena e nem organizada, você termina com arquivos dispersos sobre todo seu HD. Significa que você tem que alocar um diretório à instalação. Significa que você não pode coexistir com uma outra distribuição Linux instalada. Significa que aqueles arquivos dispersos não são carregados "do início", "não estão frescos" a cada novo boot e sendo assim são suscetíveis ao "system rot" (decomposição do sistema).

Assim, o Tiny Core é projetado para começar pequeno e ir se transformando em um sistema baseado em suas próprias necessidades através de nosso repositório da extensão. Isto difere de instalar um sistema básico Debian e adicionar programas. Começando com uma instalação Debian básica, isso significa alocar espaço no drive, não ser capaz de coexis-

**Nós do Tiny Core tentamos encontrar o contrapeso entre o tamanho e desempenho.**

Robert Shingledecker

tir, adicionando pacotes que serão espalhados sobre o sistema de arquivos. Isso é muito diferente. Considere também o que as dependências da instalações do Debian ditariam um tamanho que seria muito maior do que o Tiny Core e seu modelo otimizado de extensão. Uma instalação base Debian traduziria em mais recursos para funcionar e não é para ser dinâmica em cada boot. O Tiny Core em qualquer de suas diversos modos de execução pode facilmente ser seu desktop. "Eu como minha própria dog food". Meu desktop é Tiny Core. Eu o rodo com umas 60 extensões. Prefiro pessoalmente o tipo de montagem do tcz. Eu tenho dois netbooks ambos têm SSDs pequenos. Uso a instalação frugal para ele coexista com o Xandros no ee-ePC 900A e com o Ubuntu no Dell 9 Mini. Eu sempre entro pelo Tiny Core. A inicialização, mesmo com o carregamento das extensões, me fornece um netbook com redes sem fio muito mais responsivo do que o OS nativamente instalado. O Tiny Core também funciona bem com computadores velhos, naturalmente, as características modernas como GTK2, o flash mais recente têm exigências enormes. Na Conferência Scale 7x que aconteceu no sul da Califórnia, eu demonstrava o Tiny Core em alguns dos netbooks mais novos tão bem quanto laptops com 300Mhz e 128MB em modo framebuffer. Todos estavam funcionando

com rede sem fio. O Tiny Core tem suporte a adição de usuários mas principalmente para permitir o acesso SSH. O Tiny Core está ainda em sua infância. Nossa primeira versão pública saiu no dia 1º de dezembro de 2008. Muitas características e atualizações ainda estão por vir.

**REL: Muitos acreditam que o DSL é uma mini-distribuição. Bem, temos agora o Tiny Core que é ainda menor. Existe possibilidade de se ter algo ainda menor do que o Tiny Core e com a mesma usabilidade? Qual é o limite?**

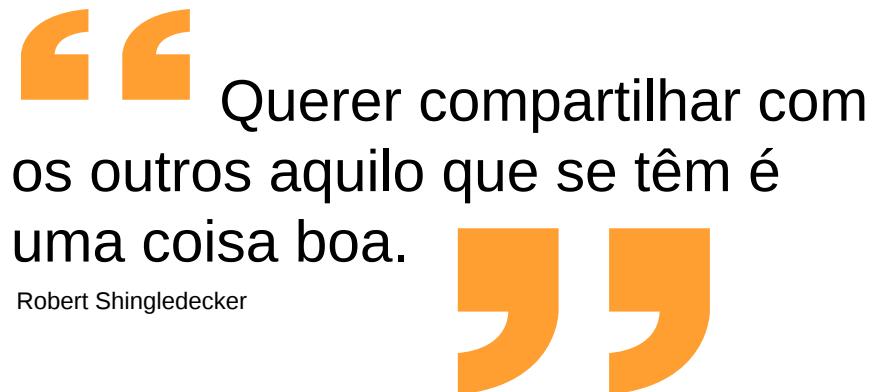
RS: O Tiny Core podia usar uma compressão mais elevada, como LZMA. Mas significaria um boot mais lento. Como resultado teríamos um menor tamanho, mas que não é tão primordial quanto o desempenho. Nós no Tiny Core tentamos encontrar o contrapeso entre tamanho e desempenho. O Tiny Core podia também usar uma pequena bi-

blioteca em C. Mas outra vez, o glibc padrão, foi usado de modo que o Tiny Core fosse mais "main stream" do que uma distribuição embarcada, especializada de Linux. O Tiny Core poderia liberar uma versão sem o X (interface gráfica) somente para ser ainda menor, porém uma versão só em linha de comando não seria muito amigável. Eu acho que me esforcei para o contrapeso correto da facilidade de utilização, tamanho e desempenho.

**REL: Você recomenda o uso do Damn Small Linux e do Tiny Core aos usuários iniciantes, que acabaram de chegar ao mundo do Linux? Você acha que são difíceis para o usuário que acabou de chegar no Linux?**

RS: O DSL é um desktop completo e por causa disso é útil aos novatos. O DSL é completo e às vezes ainda é bom tamanho para um hardware mais antigo.

Pode-se dizer que o Tiny Core é para usuários avançados



Querer compartilhar com os outros aquilo que se têm é uma coisa boa.

Robert Shingledecker

dos. Mas eu tenho tentado duramente apresentar uma interface fácil de se usar em relação a adicionar aplicações, módulos, e bibliotecas. Sendo somente um núcleo, isso implica em encarar muitas escolhas. Com muitas escolhas, vêm muitas decisões. Para tomar decisões eficazes temos que ter tempo para aprender tudo que envolve o Tiny Core. Tente você mesmo, você pode ficar muito surpreso como pode ser fácil.

**REL: Que você pensa dos remixes que muitos fazem com distribuições Linux? É saudável?**

RS: O software livre promove muito isso. Eu penso que é saudável. Qualquer um que queira aprender os fundamentos de como o Linux funciona é algo muito bom. Querer compartilhar com os outros aquilo que se têm é uma coisa boa. Muitos remixes podem não ser muito interessantes, mas fazer o que?! Sem nunca ter a oportunidade, como é que novas visões iriam

acontecer?

**REL: Poderemos encontrar no futuro o Damn Small Linux ou o Tiny Core em uma PC à venda em uma loja grande, (de departamentos ou shopping) com o sistema instalado / OEM?**

RS: Eu acho que não. O DSL é muito atual. O Tiny Core não é um desktop completo. O Tiny Core está mais para hobbystas, construtores do sistema, construtores do dispositivos, etc. O Tiny Core não é uma solução instantânea. Agora, um construtor/integrador de sistema pode desdobrar e muito o Tiny Core nos dispositivos que eu nem mesmo saberia. Recentemente, me mandaram um email mostrando que o DSL estava sendo usado no "Bay Area Rapid Transit (BART)" de São Francisco, Califórnia para indicar a informação do trem. Em uma conferência recente de Linux, se aproximaram de mim para dizer que meu software estava

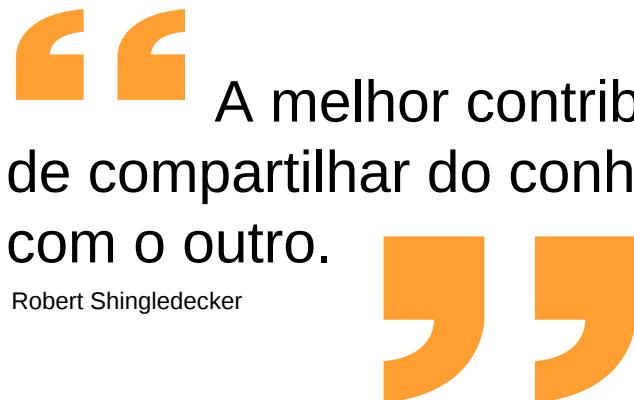
sendo usado anunciar a informação do estacionamento em bondes no "Los Angeles Fairgrounds".

**REL: Como a distribuição Tiny Core é sustentada? Como sobrevivem?**

RS: O Tiny Core é mantido por uma equipe dos voluntários (<http://tinycorelinux.com/Teram TC.html>). Por causa da minha idade e saúde, eu me habilito a fazer menos a cada ano. Mas não tenho medo algum, a equipe que eu selecionei é mais do que capaz. Mas eu não vou desistir. Desde que eu seja capaz, estarei dirigindo o desenvolvimento do Tiny Core.

**REL: Atualmente você está a frente do Tiny Core. Como ficou sua participação no DSL? Você tem outros projetos de código-fonte aberto?**

RS: Desde que eu comecei o Tiny Core, John Andrews proibiu-me de continuar a trabalhar no DSL. Fundou o DSL, possui a marca registrada, e controla todo o acesso ao DSL. Não tenho nenhum outro projeto de código fonte aberto. Desde que fiquei aposentado, eu vivo da minha pensão e investimentos. Nós, eu e equipe, somos todos voluntários. Os conceitos Tiny Core são algo que todos acreditam. É nossa

A melhor contribuição é a de compartilhar do conhecimento com o outro.  
Robert Shingledecker

paixão.

**REL:** Quando não está atrás de um computador no desenvolvimento do Tiny Core, que faz para se divertir? Você trabalha 24 horas?

RS: Já que sou deficiente, eu aprendi a usar o transporte público. Eu conheço todos os ônibus, trilhos, e sistemas de trem. Ter OPMD significa muita dificuldade para engolir, eu tenho necessidades especiais em relação à comida. Eu encontrei muitas comidas, de diferentes culturas, interessantes e saborosas que são fáceis de engolir. Tendo aprendido tudo isso, planejo agora "buscas" para tentar alimentos de muitas culturas. Eu conduzo um grupo pequeno de outros cidadãos na terceira idade nestes lugares. Nós nos divertimos muito e aprendemos provando alimentos de muitas culturas. A autenticidade verdadeira, étnica, a variedade nativa, do Afeganistão, da África, do Peru, do sudeste da Índia, da Coréia, da China, do México, etc.

**REL:** Como você vê o crescimento de Linux em computadores atuais? Existe a possibilidade de tornar-se ainda mais popular?

RS: Com mais dispositivos vêm mais oportunidades de inovação. Linux e seu mode-

lo de fonte aberta oferecem mais oportunidades do que o mundo de fonte fechado. Assim, penso que o futuro é muito brilhante para o Linux. Ainda é muito difícil tornar-se em um negócio grande. É tipicamente controlar o que está prontamente disponível no mercado, e como tais controles, o estado mental que isso provoca nas pessoas.

**REL:** Onde nós podemos aprender mais sobre o DSL e o Tiny Core? Que você recomenda? Que Web site ou livros?

RS: Até onde tenho conhecimento, o DSL, ainda enquanto um desktop útil, é um projeto terminado e fechado. A documentação definitiva para ele é "O Livro Oficial Damn Small Linux", publicado por Prentice Hall, Pearson Education. Já para o Tiny Core, eu espero que vocês venham nos visitar em <http://tinycorelinux.com>. Nós não temos nenhum anúncio, não pedimos nenhum dinheiro. O Tiny Core Linux é sobre compartilhar, aprender, e promover um divertimento novo e de uma maneira emocionante de usar o Linux. A melhor contribuição é a de compartilhar do conhecimento com o outro.

**REL:** O que os fãs e os novos adeptos do Tiny Core (e do DSL) podem esperar das versões seguintes des-

tas distros?

RS: Desde que fui exilado do projeto, eu não posso responder sobre o futuro do DSL.

O que é emocionante sobre o Tiny Core é o modelo de extensão. Através das extensões nós podemos continuar em qualquer direção. Nós manteremos o núcleo atual e nos esforçaremos por mantê-lo pequeno. Adicionaremos uma infra-estrutura como necessidade para melhorar o suporte e a interação com nossa modelo de extensão. Somos uma distribuição nova e estamos apenas começando. O próximo release incluirá maiores atualizações do kernel e bibliotecas do núcleo.

**REL:** A Revista Espírito Livre lhe agradece pela oportunidade! Deixe uma mensagem para nossos leitores.

RS: O Tiny Core é divertimento. É desafio. Tem muitas possibilidades. Eu espero que vocês visitem, explorem, e partilhem.

Muito obrigado por proporcionar-me esta oportunidade de compartilhar de minha paixão com os leitores. 

## Maiores informações:

**Site Tiny Core Linux:**

<http://www.tinycorelinux.com>

**Site Pessoal:**

<http://www.shingledecker.org>

# ENTREVISTA: FLÁVIO DE OLIVEIRA Criador do GoblinX

Por João Fernando Costa Júnior

A edição n. 2 da Revista Espírito Livre traz em sua capa um assunto bem interessante e que nos reporta a vários sub-assuntos. Em meio as diversas tecnologias que podem ser utilizadas visando a utilização de máquinas leves e mais rápidas encontramos distribuições que visam todos estes atributos, mas com um "q" a mais de beleza. Assim, entrevistamos Flávio de Oliveira, responsável pela distribuição GoblinX, uma distribuição que merece atenção.

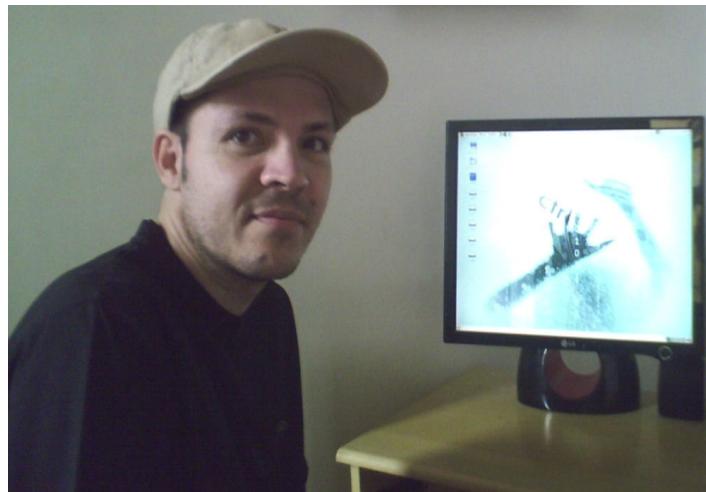


Figura 1: Criador e criatura

**Revista Espírito Livre:  
Quem é Flávio de Oliveira?  
Fale-nos um pouco sobre você.  
Quando começou com sua história com os computadores e software livre? E nas horas vagas, o que faz para se divertir?**

Flávio de Oliveira: Eu sou engenheiro civil de formação e sempre utilizei computadores porque trabalhava com projeto civil, preparando plantas baixas, orçamentos e especificações. Sou ainda artista plástico e escritor de forma amado-  
ra ainda.

A história com o software livre começou quando eu passei a ter internet banda larga. No início comecei trocando meus programas por outros livres,

por exemplo eu já usava Abiword no Windows, até que um dia resolvi migrar tudo para software livre e conheci o Linux.

**REL: Como surgiu a idéia de criar o GoblinX? O que te motivou? E de onde vem o nome?**

FdO: Eu comecei usando Slackware e um dia enquanto pesquisava por jogos para utilizar encontrei o Kurumin Games, foi então que percebi a possibilidade que todos tinham de remasterizar e criar uma distribuição, assim nasceu a idéia de fazer meu próprio livecd. O nome GoblinX veio meio do nada, sem muitas explicações, goblin é um termo utilizado em praticamente todas as línguas, é meio um bandido no mundo das lendas e traz esse jeito indiferente, malandro que gostaria de dar ao projeto, adicionei o X no fim apenas.

**REL:** Pensamos que o termo beleza e leveza descreve bem o GoblinX. Quais são as particularidades do sistema diante das demais distribuições linux nacionais e internacionais? O que o torna diferente dos outros?

FdO: Beleza e leveza não são mais as principais características do GoblinX nem as maiores diferenças com as co-irmãs. Durante um tempo, no início, beleza e customização foram fundamentais, contudo hoje o GoblinX traz como maior diferença entre as principais distribuições ser a única, em diria em conjunto com o Zenwalk, a oferecer o melhor sistema para livecds disparado, o Linuxlive, com recursos após instalação no padrão das gigantes, como repositório próprio de pacotes. Esse é o diferencial, a união entre o melhor sistema para livecds com o necessário para se tornar uma ótima distro instalada. Não tem

como comparar um livecd do Ubuntu com o GoblinX, o GoblinX é muito superior, então tendo essa força no livecd o que procuramos é tornar o GoblinX instalado um sistema com recursos dos grandes sistemas.

**REL:** O GoblinX tem várias versões. Qual a diferença entre elas? Você não acha que tantas versões podem confundir aquele que não sabe por onde começar?

FdO: Todos os sistemas operacionais grandes tem diferentes versões, incluindo Windows e Ubuntu. No caso do GoblinX a idéia é aproveitar ao máximo o poder do Linuxlive, virtualmente eu poderia montar milhares de sistema diferentes com os recursos atuais, atualmente passei a fazer quatro distros, uma para cada um dos desktops principais, Kde, Gnome, Xfce e Fluxbox, aproveitando o melhor

de cada um destes, oferecendo aos usuários mais opções e ainda recebendo mais ajuda para melhorar os ambientes.

**REL:** Particularmente no quesito leveza, o que você tem a dizer sobre o

**GoblinX? Quais os requerimentos mínimos para se usar o GoblinX?**

FdO: A leveza entra na pergunta acima também, por termos diferentes distribuições, temos diferentes possibilidades. O G.Mini e o G:Micro que utilizam respectivamente Xfce e Fluxbox são indicados para micros mais抗igos, aliás o meu micro de trabalho é antigo, um Athlon 2600+ com 512MB e rodo as quatro distros, claro que KDE e Gnome são mais pesados. O G.Mini e G.Micro podem ser usados bem em micros piores, mas tudo depende do que o usuário irá rodar, alguns programas exigem mais do que outros. O G.Mini e G.Micro eu testo em um Pentium III de 256kb normalmente.

**REL:** Quando o linux nasceu era tido como um sistema que consumia poucos recursos e baixo poder de processamento. Conforme o sistema foi amadurecendo a exigência por um hardware cada vez melhor também aumentou. Hoje temos distribuições com vários cds e dvds, tornando-se um sistema completo, e em contra-partida pesado. O que você pensa disso? Na sua opinião os desenvolvedores devem se preocupar com a exigência de hardware ou tanto faz, haja vista que o valor das peças sempre tende a cair?

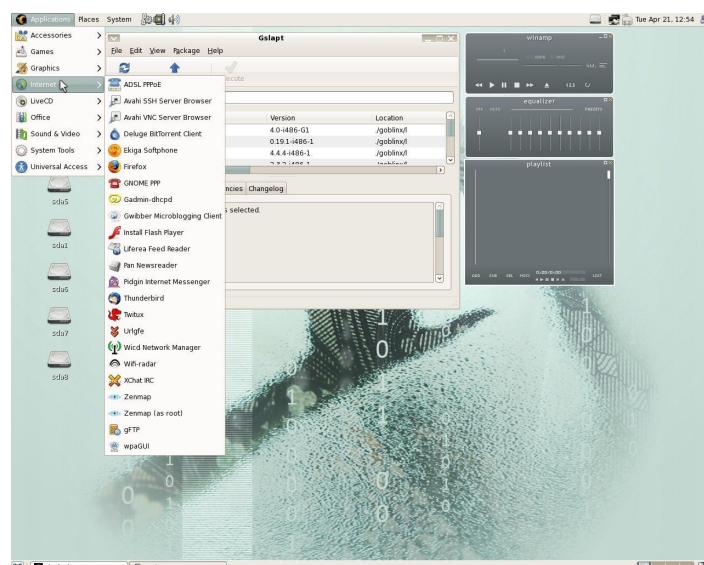


Figura 2: Tela do GoblinX em execução

FdO: Acho que devem existir distribuições e projetos pensados para micros mais抗igos, contudo os maiores projetos como Kde e Gnome devem focar usabilidade e correção de bugs e não leveza. Afinal existem outros programas e gerentes de janelas que pensam em leveza. Openoffice.org, Firefox e outros poderosos aplicativos devem tentar ser o mais leve possível mas nunca retirar ou deixar de adicionar recursos por causa de poder de processamento.

**REL:** Vemos nascer a cada dia novas distribuições linux baseadas em outras. Ao mesmo tempo vemos várias morrerem por falta de suporte e comunidade. O GoblinX é baseado em alguma distribuição? O que você acha das distribuições que só não tem autenticidade e só basicamente trocam temas e são lançadas novamente como uma nova distribuição?

FdO: O GoblinX é baseado no Slackware. Ele utiliza o Linuxlive na confecção do livecd, então o GoblinX é bastante diferente se comparado ao Slackware, que nem é livecd. Acho que uma distribuição deveria nascer seguindo a GPL, não precisa ter recursos novos e sim dedicação. Com o tempo a distribuição irá crescer e com ela a comunidade, nenhuma distribuição nasce grande e sou contra essa idéia que muitas distribuições atrapalham.

Contudo existe um outro lado, o da pura e simples personalização, não acho viável nem importante e estas distros acabam morrendo, mas cada um faz o que quer, democracia é um dos pontos do mundo de software livre. Se um indivíduo acha que a customização dele é interessante e ele segue as regras da GPL, deve lançar.

**REL:** Não sei se vocês tem essa informação, mas como o GoblinX é visto fora do Brasil? Qual a repercussão? Você tem números quanto a número de usuários no Brasil e fora dele?

FdO: Fora do Brasil o GoblinX é bem mais conhecido chegando a estar entre as trinta e cinco primeiras no [Distrowatch.com](http://Distrowatch.com) no passado. Os números de usuários já pos o Brasil em quarto lugar na lista, atrás de Estados Unidos, Alemanha e França, hoje os usuários brasileiros aumentaram mas ainda estão em número bem menor que os americanos, o que é natural afinal nos Estados Unidos existem muito mais usuários de computador.

**REL:** Na sua opinião o



Figura 3: Interface Netbook

**GoblinX contempla o usuário leigo ou é um sistema para os mais entendidos de linux? A quem ele é destinado?**

FdO: Eu diria que é um sistema entre fácil e intermediário, com um ponto fundamental, o poder do usuário é levado em conta, penso sempre em incluir facilidades sem tirar o poder do usuário de alterar tudo.

**REL:** Hoje no Brasil encontramos inúmeras distribuições. Entretanto as duas mais populares (Conectiva e Kurumin) tiveram nascimento e morte. Com isso alguns usuários ficaram "órfãos" por assim dizer. Esta falta de garantia que o sistema continuará tendo continuidade coloca medo em algumas pessoas. O que você diria para eles?

FdO: O GoblinX seguirá um caminho inverso, o da profissionalização, a distribuição irá crescer, ser reformulada e entrar em um outro patamar.

**REL:** Você acredita que o cenário nacional tem lugar para as tantas distribuições que tem? Se as comunidades se unissem será que não teríamos algo mais sólido e enbasado? No mundo temos mais de 300 distribuições. Aos olhos de quem está chegando parece que cada um está atirando para um lado. O que você pensa disso?

FdO: Em relação a união entre comunidades eu não acredito que traria melhoria alguma, cada comunidade deve ajudar e ter opção de um modo geral, acho mais importantes as comunidades dos projetos do que das distribuições, só que as comunidades em volta de uma distribuição tendem a ter mais força.

Agora eu acho que quem atrapalha são as grandes empresas, Canonical, Novel e Red Hat cada qual faz algo levando em conta apenas as idéias deles e sem considerar os outros, assim todo dia o Ubuntu traz uma novidade, o Fedora outra e o SuSE outra e estas não se ajudam em quase nada e normalmente estas novidades fazem a mesma coisa que outros projetos já faziam.

Nós temos poucas distribuições no Brasil, já tivemos mais.

**REL:** Na sua opinião o que falta em um sistema linux hoje para que ele assu-

ma de vez uma posição no mercado, com chances reais de desbancar concorrentes (se é que podemos chamar de concorrentes)?

FdO: Acho que as grandes cooperações precisam se ajudar mais, Canonical e Novel por exemplo parecem brigar entre elas, acho que falta ainda mais parcerias dessas empresas com grandes fabricantes de software, como Adobe, Autodesk e outras mais. Falta ao Linux ser compatível com mais programas dos grandes fabricantes como Adobe, ser mais maleável e aceitar aplicativos e drivers não livre e trabalhar mais os nomes das distribuições do que a marca Linux.

**REL:** Com a chegada de sistemas operacionais proprietários que exigem bastante hardware, o preço sempre tende a cair. Você acha que com essa medida os beneficiados somos nós que utilizamos linux?

FdO: O Linux sempre se beneficia das mudanças nos grandes sistemas, entretanto isso é passageiro, não acho que aquele que testa e odeia o Windows Vista adote alguma distro Linux, é mais fácil retornar ao Windows XP. O Linux se beneficiaria muito mais se o suporte ao Windows XP dado pelos grandes fabricantes de aplicativos e hardware fosse interrompido.

**REL:** O que os fãs da distribuição e colaboradores podem esperar nas futuras versões do sistema?

FdO: O GoblinX vem sempre mudando e renascendo a cada versão, e para a próxima passaremos por nossa maior mudança, com um novo modo de gerenciar o projeto para torná-lo profissional e entrarmos no mercado dos fabricantes de computadores.

**REL:** Caso alguém queira colaborar com o sistema o que deve fazer? E onde o usuário leigo, que acabou de ficar sabendo do sistema, deve buscar informação e dicas?

FdO: O melhor modo é me contactar via email contato@goblinx.com.br ou visitar nosso site e fórum. O site e fórum tem muitas dicas, artigos e sempre eu ou alguém tirando as dúvidas.

**REL:** Deixe uma mensagem para o pessoal que está lendo a revista!

FdO: Espero que você leitor tenha gostado da entrevista e se lembre de falar com amigos sobre o GoblinX e sobre a Revista Espírito Livre.

### **Maiores informações:**

**Site Oficial GoblinX Linux:**  
<http://www.goblinx.com.br>



Por Aécio Pires

O que fazer quando os desktops estão obsoletos demais para usar a versão mais nova de um sistema operacional, por exemplo? O que fazer para reduzir os custos de propriedade da rede (custos com a aquisição e manutenção dos equipamentos, licença de uso de softwares proprietários, consumo de energia elétrica, etc)? Ou o que fazer para reduzir o impacto ambiental dos computadores obsoletos?

Uma boa solução para esses e outros problemas é a utilização de clientes magros. Uma rede de clientes magros possui baixo custo de instalação por permitir o uso computadores obsoletos ou thin clients[1], todos conectados a um servidor de alta performance de processamento que compartilha o sistema operacional, aplicativos, acesso a Internet, CD-ROM, impressora, armazenamento de arquivos, etc.

Existem vários projetos de Software Livre e Proprietário que proporcionam a implantação dessas redes, mas neste artigo iremos conhecer o TCOS – Thin Client Operating System.

## O que é TCOS?

É um projeto de Software Livre licenciado pela GPL 2[2] e formado por um conjunto de ferramentas gráficas utilizadas tanto para inicializar como para gerenciar clientes magros.

O TCOS foi criado pelo espanhol Mario Izquierdo Rodrigues e encontra-se disponível no site <http://tcosproject.org> para ser instalado nas

distribuições GNU/Linux Debian[3], Ubuntu[4], MaX 4.0[5], Guadalinex[6] e Lliruex[7]. A versão atual é a 0.89.33 e está traduzida para o espanhol, inglês e português.

## As ferramentas

Os grandes trunfos do TCOS, em relação a outros projetos de Software Livre para clientes magros, são as ferramentas gráficas:

**tcosconfig** => usada para criar e personalizar os arquivos de inicialização dos clientes. Desenvolvida em Python e GTK2, ela é uma interface gráfica para o gentcos, um shell script que compila as imagens a partir das configurações indicadas pelo administrador.



Figura 1: Tela inicial do TcosConfig

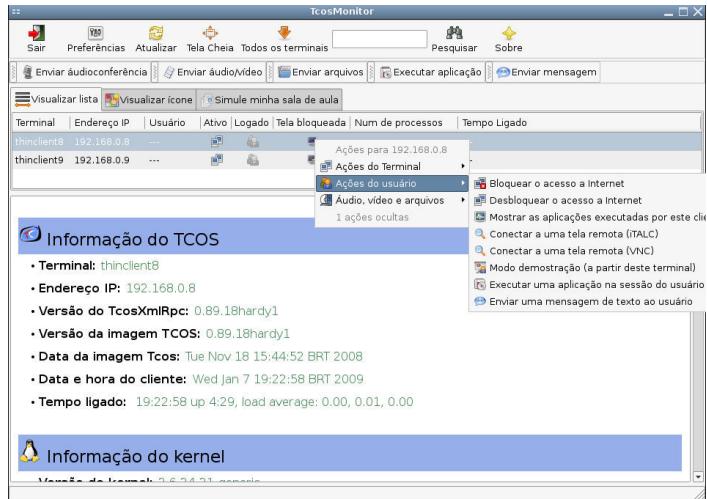


Figura 2: Tela do TcosMonitor em uso

**tcosmonitor** => utilizada para gerenciar os clientes magros e desktops, desde que tenham instalado a aplicação cliente tcos-standalone. Com o tcosmonitor é possível reiniciar/desligar os clientes, bloquear ou desbloquear a tela, teclado e mouse; controlar o cliente utilizando o VNC, capturar a tela, encerrar a sessão gráfica dos usuários, executar uma aplicação remota, enviar mensagens aos usuários conectados, visualizar/encerrar as aplicações em execução, acessar as informações de cada cliente (o uso do processador, consumo de memória RAM, módulos do kernel em uso, configurações da interface de rede, processos em

execução, etc), compartilhar arquivos de multimídia, realizar audioconferência, entre outras.

**tcosphpmonitor** => uma alternativa ao tcosmonitor permitindo gerenciar os clientes através de um sistema Web, desenvolvido em PHP.

**tcospersonalize** => permite configurar algumas características dos clientes, tais como: a resolução da tela, os drivers de vídeo, módulos de kernel, entre outros.

**tcos-device-ng** => utilizada para monitor/desmontar automaticamente o CD-ROM, disco rígido, dispositivos USB, entre outros dispositivos conectados nos clientes.

**pam-usb-tcos** => um utilitário gráfico que associa o uso de um dispositivo USB a um ou mais usuários. Ele evita que o usuário digite a senha para montar ou desmontar o dispositivo todas as vezes em que ele for conectado ou desconectado nos clientes.

**tcos-configurator** => permite ao administrador configurar o servidor TCOS e alguns serviços de rede relacionados, como: o DHCP, cadastro de usuários no sistema e configuração do gerenciador de login.

## TCOS x LTSP5

Talvez você esteja se perguntando: TCOS não é igual ao LTSP[8]? Ambos tem os mesmos objetivos? Quais as diferenças entre eles?

Segundo o manual do TCOS para administradores[9], escrito pelo desenvolvedor do TCOS, esses dois projetos possuem os mesmos objetivos, mas as diferenças estão na implementação de cada um. Eis algumas das diferenças existentes entre o TCOS e o LTSP, versão 5:

- No LTSP 5 os clientes usam NFS (Network File System) para obter o sistema de arquivos raiz (opcionalmente podem usar squashfs sobre NBD - Network Block Device).

Figura 3: Tela do TcosPersonalize em uso

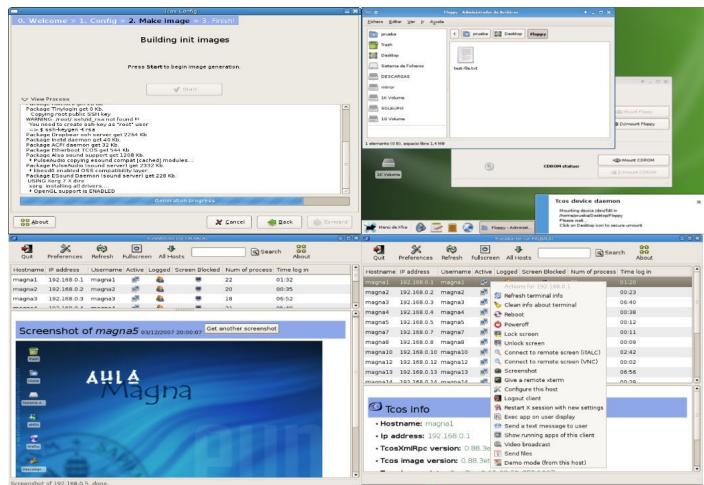


Figura 4: Ferramentas do Tcos

No TCOS, os clientes obtém a imagem squashfs usando os protocolos TFTP (Trivial File Transfer Protocol), HTTP (Hipertext Transfer Protocol) ou NFS, mas quando o cliente possui mais do que 38 MB de memória RAM não é necessário utilizar NFS, o que diminui o consumo de memória RAM.

- Se o servidor LTSP 5 falhar ou perder conectividade, todos os clientes sofrerão um kernel panic e precisarão ser reiniciados. Com o TCOS, eles apenas tem de esperar o servidor voltar a funcionar normalmente.

- O sistema de acesso aos dispositivos removíveis utilizado por ambos os projetos é o LTSPFS (filesystem sobre Xorg + o módulo de kernel chamado FUSE). No TCOS houve uma melhoria no uso desses dispositivos com a aplicação gráfica tcos-devices-ng. No LTSP 5 esses dispositivos são montados de forma automática e a desmontagem é complicada, mas no TCOS essas operações são realizadas de forma muito simples.

- LTSP 5 só tem suporte a disquetes, memórias USB e CD-ROM de dados. O TCOS, além desses dispositivos, tem suporte a partições do disco rígido (incluindo o sistema de arquivos NTFS, CD-ROM USB, discos firewire, CD de áudio, etc).

- No LTSP 5 é necessário ter uma conexão

com a Internet, um repositório local ou um CD-ROM/DVD-ROM com os pacotes para criar a imagem de inicialização dos clientes. TCOS utiliza os binários disponíveis no servidor. No LTSP5 esse processo demora mais de 15 minutos, no TCOS apenas 15 segundos.

- LTSP 5 não permite personalizar a imagem de inicialização dos clientes. Com o TCOS é possível personalizá-la e escolher o que será removido ou incluído.

- LTSP 5 tem uma ferramenta muito simples para administrar os clientes chamada Thin Client Manager e outra para configurá-los, a LTSP-Manager. TCOS possui as ferramentas tcosmonitor e tcospfmonitor, para gerenciar os clientes, e a tcospersonalize, para configurá-los.

- LTSP 5 só permite conexões gráficas entre os clientes e o servidor usando o XDMCP (X Display Manager Control Protocol). Além desse protocolo, o TCOS utiliza: o rDesktop (para Windows Terminal Server), FreeNX, SSH+X e o XRDP.

- TCOS usa o XMLRPC entre o servidor e os clientes para trocar informações, tornando mais fácil o desenvolvimento de pequenos programas ou plugins para fazer tarefas simples, como por exemplo pam-usb-tcos.

- A configuração do servidor LTSP 5 é feita através da edição manual de arquivos de configuração. Além desse método o servidor TCOS pode ser configurado usando a aplicação gráfica tcos-configure, tornando esse processo rápido e fácil.

- LTSP 5 não tem uma aplicação gráfica para configurar e compilar as imagens dos clientes. TCOS possui o tcosconfig.

Por outro lado, o LTSP é um projeto que existe a mais tempo, sendo bem mais conhecido e utilizado. O LTSP também possui versões para outras distribuições GNU/Linux, como: Slackware[10], Fedora[11], etc.

## Quem está usando TCOS?

- Conheça alguns dos casos de sucesso do TCOS no mundo:
- Escola Universitária Politécnica de Valladoid (Espanha);
- Colégio La Salle - Felipe Benito (Servilha, Espanha);
- Colégio El Apostolado (Valladolid, Espanha);
- Escola de Ofícios San Cayetano Centro de Formación Profesional de Vicente López Carapachay (Buenos Aires, Argentina);
- Rádio Comunitária Sur FM 102,7 (Buenos Aires, Argentina);
- Universidad Yacambú (Barquisimeto, Venezuela);
- Biblioteca do Liceo Industrial de Santiago (Santiago, Chile).

No Brasil, ainda não foi registrado oficialmente nenhum caso de sucesso.

O TCOS também foi vencedor do I Concurso Universitário de Software Livre, promovido pela Universidade de Servillha-ES, em 2007, na categoria “Distribuições” (<http://concurso-software-libre.us.es/0607/>).

## Considerações Finais

Então, o que você achou do TCOS? Se estiver curioso acesse o site oficial do projeto[12] e da comunidade TCOS Brasil[13], ambos possuem uma vasta documentação sobre TCOS. ↗

**Este artigo é o primeiro da série e nos próximos iremos entender o funcionamento de uma rede TCOS, os protocolos utilizados, as configurações de hardware recomendadas para os clientes e o servidor, o impacto que o TCOS causa numa rede de desktops, conhecer as vulnerabilidades e as contra-medidas para tornar a rede mais segura. Não percam!**

## Maiores informações:

### [1] Thin Clients

[http://pt.wikipedia.org/wiki/Thin\\_client](http://pt.wikipedia.org/wiki/Thin_client)

### [2] GPL 2

<http://www.gnu.org/licenses/gpl-2.0.html>

### [3] Debian

<http://debian.org/>

### [4] Ubuntu

<http://www.ubuntu.com/>

### [5] MaX 4.0

[http://www.educa.madrid.org/web/madrid\\_linux/](http://www.educa.madrid.org/web/madrid_linux/)

### [6] Guadalnex

<http://www.guadalnex.org/>

### [7] Lliruex

<http://lliruex.net/>

### [8] LTSP

<http://www.ltsp.org/>

### [9] Manual do TCOS para Admins

[http://br.tcoproject.org/?page\\_id=16](http://br.tcoproject.org/?page_id=16)

### [10] Slackware

[http://www.slackware-brasil.com.br/web\\_site/](http://www.slackware-brasil.com.br/web_site/)

### [11] Fedora

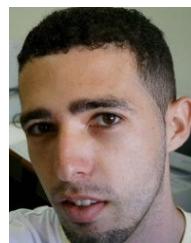
<http://www.projetofedora.org/>

### [12] TCOS Project

<http://tcoproject.org/>

### [13] TCOS Brasil

<http://br.tcoproject.org/>



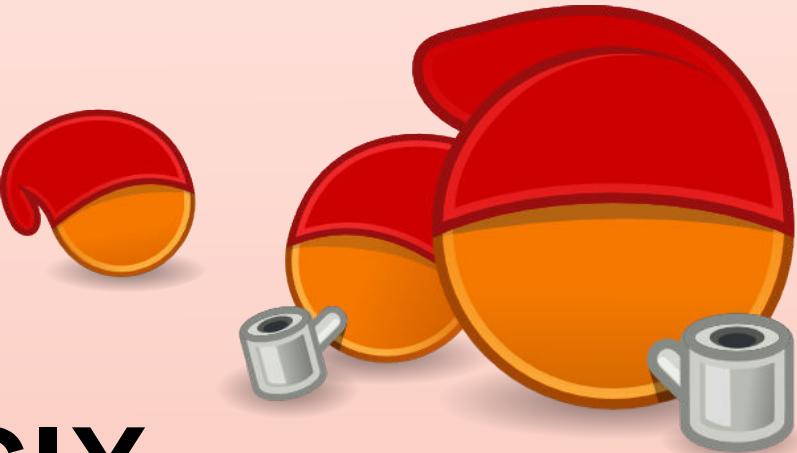
**AÉCIO PIRES** (<http://aeciopires.rg3.net>) é graduando em redes de computadores pelo IFPB ([www.ifpb.edu.br](http://www.ifpb.edu.br)), tradutor do TCOS, fundador da comunidade TCOS Brasil e estagiário da Secretaria da Receita do Estado da Paraíba, onde faz pesquisas com TCOS desde junho de 2008.

# TERMINAIS LEVES COM O SACIX

Por Anderson Goulart e Luiz Paulo

Terminals leves, thin clients ou ambiente diskless (sem disco) são praticamente sinônimos nos textos técnicos sobre o assunto. O "praticamente" vem do fato de um sistema leve poder ter discos rígidos, mas ainda ser encarado como uma solução para evitar o sucateamento dos computadores. Neste aspecto, surgiram várias soluções comerciais e livres que são capazes de reviver computadores com baixo poder de processamento e memória. Os 2 grandes projetos livres relacionados são o LTSP (Linux Terminal Server Project)[1] e o DRBL (Diskless Remote Boot in Linux)[2]. Ambos provêem um terminal remoto, mas suas construções internas são diferentes. A idéia é simples: coloca-se um computador de alto desempenho servindo os outros, onde todo o processamento é realizado. Durante a execução dos programas, as informações sobre o que deve ser exibido na tela é enviada pela rede para cada computador conectado a ele. Dessa forma, um 486 poderia ser quase tão rápido quanto um processador de núcleo duplo, na teoria. Essa é a base para o não sucateamento dos equipamentos抗igos.

A prática revela que vários são os gargalos ou problemas enfrentados por essas soluções: infra-estrutura de rede, memória de vídeo e RAM do cliente, suporte aos dispositivos locais (portas usb, pendrives, placas de som, impressoras) e o desempenho geral do sistema quando se ten-



ta escalar a solução. Entretanto, em pequenas proporções, é possível criar um ambiente útil de acesso à internet e às aplicações convencionais (editores de texto, planilhas, mensageiros, etc), com um custo reduzido de infra-estrutura, aproveitando os computadores sucateados.

Com o objetivo de facilitar instalação e configuração de um ambiente baseado no LTSP criamos o Sacix[3] que, atualmente, é formado por pacotes e meta-pacotes DEB[4] capazes de automatizar esse processo. Os terminais leves são computadores convencionais, com baixo poder de processamento e geralmente sem discos rígidos. Na seção seguinte abordaremos a arquitetura e funcionamento do sistema. Este projeto é utilizado principalmente nos telecentros do Casa Brasil[5], mas é facilmente expansível para qualquer tipo de customização.

## Arquitetura do sistema

A figura 1 mostra a interconexão de rede exigida para o funcionamento do LTSP: um servidor (LTSP Server) interligado com os clientes (ou terminais) através de um switch. Este servidor pode ou não estar na internet compartilhada com os terminais (LTSP workstations).

Neste esquema, o Sacix instala e configura os seguintes serviços no servidor para o funcionamento do LTSP:

- **TFTP** (Trivial File Transfer Protocol): utilizado para transferir a imagem do kernel para cada estação
- **DHCP** (Dynamic Host Control Protocol): fornece os endereços IPs e nomes para cada cliente
- **NFS** (Network File System): compartilha os arquivos e diretórios utilizados pelos clientes
- **XDMCP** (X Display Manager Control Protocol): habilita o suporte para execuções de aplicações gráficas remotamente
- **LTSP**: realiza o trabalho de configuração de cada terminal e o suporte aos dispositivos: vídeo, som, impressoras, teclado, mouse, etc.

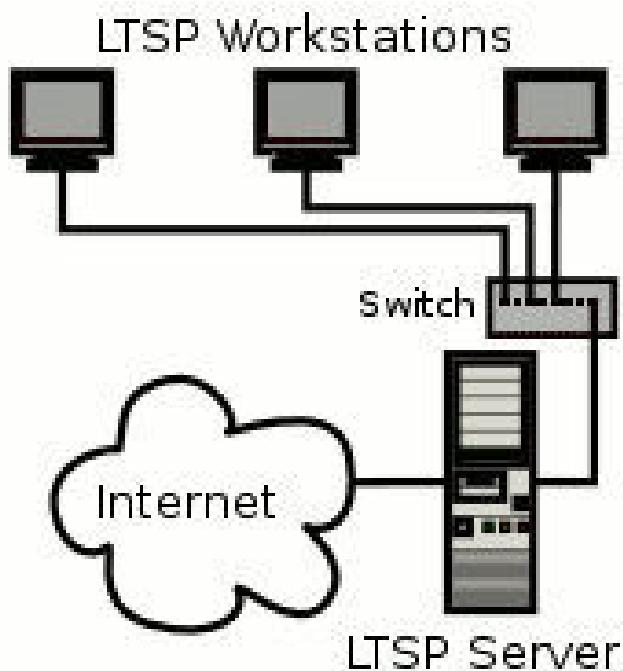


Figura 1: Arquitetura do sistema LTSP

É importante lembrar que todas as estações devem ter suporte a boot pela rede via PXE. Ou seja, todo o carregamento do sistema operacional é feito pela rede. A principal vantagem do LTSP é a facilidade de manutenção, já que administramos apenas o servidor. Não há mais necessidade de reinstalação dos clientes. Basta reiniciar a máquina. Entretanto, uma falha no servidor impede o uso de todas as estações.

O PXE pode ser habilitado na BIOS e está disponível em todas as placas-mãe modernas. Isso é um problema para os computadores anti-

gos, como os 486. Neste caso, é necessário iniciar o computador por disquete, cd-rom ou uma placa de rede externa com este suporte[6].

Já o Sacix é formado pela distribuição Debian Lenny acrescentados os pacotes de configurações dos serviços já listados. Selecioneamos os aplicativos a serem instalados a partir das demandas dos usuários dos telecentros. Em uma estrutura lógica de desenvolvimento, dividimos o nosso trabalho em 2 partes: servidor e desktop.

**Servidor (sacix-server)**: nos preocupamos com a funcionalidade do ambiente LTSP, compartilhamento da internet, segurança, serviços de impressão, proxy, compartilhamento de arquivos e configuração dos serviços.

**Desktop (sacix-desktop)**: o trabalho consiste na escolha do ambiente gráfico (Gnome), dos aplicativos (Firefox, Broffice, GIMP, Inkscape, Pidgin, etc) e suas personalizações (temas, tunning, preferências). Recursos como acessibilidade (gnome-orca) e multimídia (áudio, vídeo, flash, java, jogos) estão incluídos nesta seção, com suas devidas limitações.

### Características do sistema

Nossa proposta não é criar mais uma distribuição, com mais um nome e mais uma série de dificuldades. Manter uma distro requer tempo, dinheiro e pessoas. Por isso nossa abordagem foi criar meta-pacotes facilmente personalizados para qualquer projeto relacionado com terminais leves. Através da instalação do pacote chamado "sacix", é desencadeado um processo de instalação dos meta-pacotes "sacix-server" e "sacix-desktop" os quais, através de uma estrutura hierárquica de dependências, faz toda a mágica. Em alguns momentos é necessária a intervenção do usuário para preenchimento de informações essenciais, tais como endereço de rede e máscara. Dessa forma, aproveitamos todos pacotes oferecidos pelo projeto Debian sem a necessidade de mantê-los.

Outro ponto de destaque são as versões

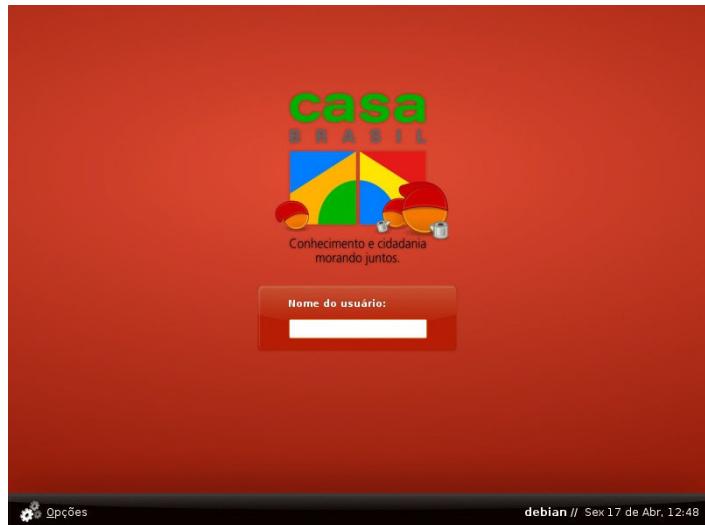


Figura 2: GDM - Tela de login do sistema

dos aplicativos. Apesar da versão Lenny não possuir as últimas versões em seus repositórios, o projeto Sacix mantém atualizado os aplicativos mais requisitados pelos usuários como Firefox 3 e Broffice 3. Assim, disponibilizamos as novas funcionalidades sendo cuidadoso para não perder a estabilidade da distribuição.

## Conclusão

Instalar e configurar um sistema com suporte a terminais leves utilizando LTSP exige um conhecimento avançado em administração de sistemas GNU/Linux. O Sacix tornou este trabalho trivial, sem perder a estabilidade e confiabilidade do Debian.

Nos telecentros, percebemos que o LTSP não substitui a instalação convencional. O uso de recursos multimídias, java, flash, jogos e suites de escritórios em vários terminais simultâneos acarreta uma grande perda de desempenho. No projeto Casa Brasil, temos um servidor (núcleo duplo, 4GB RAM) alimentando 20 estações. Mesmo neste ambiente o uso de alguns recursos são insatisfatórios. Portanto, o uso do LTSP ou outra solução relacionada é recomendado para recuperar máquinas em desuso com objetivo de torná-la um terminal leve, com recursos limitados, mas úteis para alguns propósitos específicos, como acesso à internet associada ao processo de inclusão digital.

Podemos dizer que o LTSP está estável e garante o funcionamento dos dispositivos locais (pendrive, cdrom, impressoras, etc) assim como os recursos em rede. A maior preocupação é o desempenho, haja vista a crescente demanda de processamento das aplicações web com ajax, flash e java embutidas nas páginas e recursos 3D dos jogos, o que deixa muitas vezes os usuários insatisfeitos com a solução, oriundos de uma LAN House.

Deixo aos leitores um espaço aberto para registrar quaisquer falhas ou dúvidas através do site do projeto[3]. Convidamos a todos a participar do desenvolvimento do Sacix, seja através de sugestões, suporte, traduções, documentações ou manutenção dos pacotes.

## Maiores informações:

### [1] Site Oficial LTSP:

<http://www.ltsp.org/>

### [5] Site Casa Brasil:

<http://www.casabrasil.gov.br/>

### [2] Site DRBL:

<http://drbl.sourceforge.net/>

### [6] Site projeto Etherboot:

<http://etherboot.org/>

### [3] Site Oficial do SACIX:

<http://www.sacix.org/>

### [7] Site Oficial Debian:

<http://www.debian.org/>

### [4] Artigo sobre formato Deb:

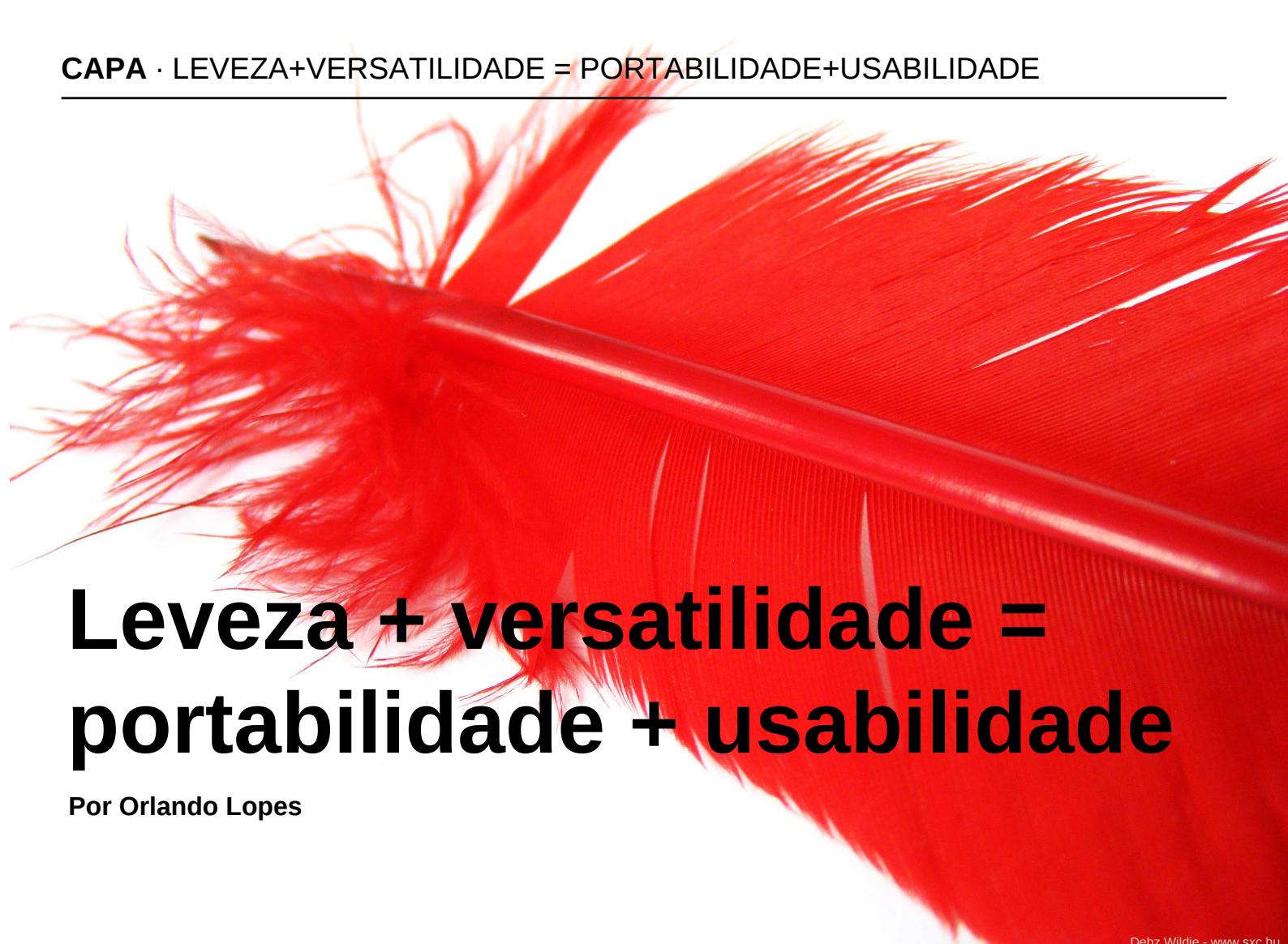
[http://en.wikipedia.org/wiki/Deb\\_\(file\\_format\)](http://en.wikipedia.org/wiki/Deb_(file_format))



**ANDERSON GOULART** é desenvolvedor do Sacix e mantenedor da infra-estrutura atual do projeto. Trabalha com inclusão digital e com sistemas baseados em terminais leves há 4 anos. É bacharel em Ciência da Computação e Técnico Especialista em Computação (TEC) do projeto Casa Brasil.



**LUIZ PAULO** atua na área de software livre há quase 10 anos. É bacharel em sistemas de informação e um dos responsáveis pelo desenvolvimento da solução Sacix para telecentros. É membro do Coletivo Digital e atualmente trabalha no data-center da empresa Cobra Tecnologia.



# Leveza + versatilidade = portabilidade + usabilidade

Por Orlando Lopes

Debz Wildie - www.sxc.hu

Há um livro de ensaios relativamente famoso de Italo Calvino (autor de, entre outros, *O barão nas árvores*, *As cosmicônicas* e *As cidades invisíveis*, livros que vale a pena conhecer), chamado *Seis propostas para o próximo milênio*. Calvino escreveu as cinco primeiras e morreu antes da conclusão da última, intitulada "Consistência". Só pra matar a curiosidade, vão aqui as outras: "Leveza", "Rapidez", "Exatidão", "Visibilidade", "Multiplicidade". Bastante adequado para quem se preocupa com as questões que se aprofundam em nossa vida "cibercotidiana", não parece?

A chamada desta edição dá atenção a aspectos bastante importantes na constituição de um sistema operacional, estabelecendo dois parâmetros que parecem bastante razoáveis para avaliar o interesse de adoção de software em geral, e do software livre em particular. Como nosso interesse aqui é buscar aspectos menos imediatos, porém também importantes para a observação crítica da concepção e implementação das redes informáticas, tentaremos apontar ao menos um desdobramento de cada um na vida individual e pública.

Em primeiro lugar, observemos que os dois termos representam características qualitativas dos recursos (sejam de hardware ou de software) a serem empregados pelos usuários. Mas por que esses dois parâmetros são importantes? Ora, eles são importantes porque correspondem a duas das principais demandas do que poderíamos chamar de “usuário modelo” das sociedades em rede. Dá até pra fazer uma equação, aquela que vem lá no título: leveza + versatilidade = portabilidade + usabilidade.

Se dizemos que um sistema deve ser leve, é porque ele não deve pesar na vida do usuário, certo? O usuário (um termo que já não seu restringe a pessoas físicas, mas também a instituições, quando falamos em sistemas) busca sempre personalizar o sistema em que opera, mas isso às vezes é penoso e muitas vezes infrutífero. Quem já não bateu cabeça procurando corrigir teclados, acertar a configuração do corrector ortográfico, reinstalar programas etc. Não será mesmo uma maravilha o dia em que nós apenas precisemos de uma senha (sim, mais uma...) para passar não apenas nossos arquivos pessoais, mas virtualmente todas as características que ele virtualmente tenha adquirido durante os “usos monitorados” (como os que se dão na escola e no trabalho)?

Hoje em dia existe já software para fazer a migração de uma máquina para outra, como o NortonGhost (proprietário) e gerenciadores de backup como o Cobian (gratuito para Windows) e o Easy Backup 0.1 (para Linux). Só não venham dizer que isso é coisa fácil de se fazer, ainda mais em se tratando de preservação de arquivos pessoais e documentos... A importância da leveza/portabilidade do “sistema de trabalho residente” no equipamento que estamos utilizando é mais que estratégica: há informações nossas que não queremos ver circulando na Nuvem, e coisas que não queremos ser obrigados a ter em nossas máquinas. Poder ir de uma máquina para outra e não perder a familiaridade de um ambiente é um dos maiores presentes que a comunidade SL poderia dar à sociedade. E ao que parece, ela vem vindo aí.

A outra questão que se desenha nos qualificativos de um bom sistema de trabalho hoje é a da “versatilidade”. Assim como as pessoas estão aprendendo a usar os computadores para atividades muito distintas na vida cibercotidiana, o sistema que elas usam deve acompanhar as variações de uso.

A Web 2.0 acelerou um processo que já vinha se desenhando, permitindo a configuração de interfaces e funcionalidades de forma bem mais intuitiva. Um computador, como todos sabemos, não é

apenas uma máquina de escrever; tudo o que entra nele – todos os inputs que fazemos nele – se reduz a um só meio, a um só veículo, o do famosíssimo “bit”. Em princípio, todos os dados podem ser representados em todos os programas, se você tiver o filtro certo. Deveria ser muito fácil – e está ficando, cada vez mais – aproveitar dados de um programa ou banco de dados em outro(s). Mas ainda não é, fazendo enorme falta e representando prejuízo de tempo na organização dos dados que dão apoio às atividades que realizamos fora do computador. 

### Maiores informações:

#### Easy Backup 0.1:

<http://ebackup.sourceforge.net>

#### Cobian Backup:

<http://www.educ.umu.se/~cobian>

#### Web 2.0:

[http://pt.wikipedia.org/wiki/Web\\_2.0](http://pt.wikipedia.org/wiki/Web_2.0)

#### Ítalo Calvino na Wikipédia:

[http://pt.wikipedia.org/wiki/Italo\\_Calvino](http://pt.wikipedia.org/wiki/Italo_Calvino)



**ORLANDO LOPES**  
é pesquisador no Núcleo de Estudos em Tecnologias de Gestão e Subjetividades do PPGAdmin/UFES e Consultor para o desenvolvimento de Projetos em Responsabilidade Cultural e Social.

# VIRADO PRA LUA

## Parte 2

Por Lázaro Reinã

Abdulaziz Almarisour - sxc.hu

### Continuando

Olá pessoal! Estamos de volta para mais um artigo sobre a linguagem Lua e sua aplicação em jogos. Espero que tenham gostado da edição anterior e que também gostem dessa. Percebiam que a partir dessa edição iremos partir para a parte mais técnica da linguagem, não mencionaremos mais toda a parte burocrática e histórica de Lua.

### Conceitos - Valores e Tipos

Lua apresenta a tipagem dinâmica. Ou seja, se temos uma variável chamada "teste" ela pode armazenar um valor de qualquer tipo a qualquer momento sem que isso seja explicitado na sua declaração. Quem conhece a linguagem C deve se lembrar do "malloc".

Nessa linguagem nós temos 8 tipos básicos, o nil, boolean, number, string, function, userdata, thread e table.

O tipo nil, que contém sempre o valor nil e é sempre diferente de qualquer outro valor, ele basicamente representa a ausência de um valor útil.

O boolean, velho conhecido da galera, representa o tipo dos valores false e true, muito utilizado em Pascal/Delphi. Retornando tanto nil quanto false torna a condição falsa, enquanto que qualquer valor diferente disso a torna verdadeira.

Number representa qualquer valor numérico, seja ele em base decimal, hexadecimal, notação científica. Ex.: 0xff; 12,5678.

String representa cadeias de caracteres, podendo conter

qualquer caractere de 8 bits, incluindo zeros ('\0').

Quanto aos demais tipos, veremos sobre eles mais à frente, quando o contexto nos permitir falar sobre eles de maneira apropriada.

### Atribuição

Como já mencionado antes, Lua tem a chamada tipagem dinâmica, ou seja, uma variável que recebe a atribuição de um dado valor passa a conter um valor de um determinado tipo, porém, caso essa mesma variável, em um momento distinto, receber a atribuição de um outro valor de tipo diferente, ela passa a conter um valor de outro tipo. É por isso que dizemos que quem tem tipo é o valor e não a variável.

A atribuição em Lua ocorre de maneira semelhante às demais linguagens, sendo representada pelo sinal “=” que diz que a variável à esquerda do sinal recebe o valor que está à direita do sinal.

Ainda é muito interessante observar uma característica no mínimo única, que se refere à atribuição de valores em Lua. É a chamada atribuição múltipla, onde podemos fazer a atribuição de vários valores diferentes à variáveis diferentes, escrevendo apenas uma linha.

```
str teste, num="  
atribuição multipla",  
10 ;
```

No exemplo acima vimos um exemplo da atribuição múltipla. Perceba que na mesma linha foi atribuído uma string para a primeira variável, e um número para a segunda variável.

## Operadores

Lua provê operadores aritméticos, relacionais, e lógicos. Além de um operador de concatenação de strings que será visto mais à frente.

### Operadores Aritméticos

Os operadores aritméticos apresentados por Lua são os usuais de qualquer linguagem.

Os binários são: Adição( + ); Subtração ( - ); Multiplicação ( \* ) e Divisão ( / ). Como

não poderia ser diferente, podemos usar parêntesis para controlar a precedência dos operadores. Vejamos alguns exemplos:

```
a = b + c  
a = (b - c) * d  
a = b / c + d
```

### Operadores relacionais

Os operadores relacionais são geralmente usados em estruturas de controle de fluxo, laços de repetição. De maneira mais generalizada os usamos quando tivermos de testar alguma expressão. São eles:

<	menor que
>	maior que
<=	menor ou igual a
>=	maior ou igual a
==	igual a
~=	diferente de

Esses operadores retornam false caso a expressão seja falsa, e 1 caso seja verdadeira. Simples, não?! Vejamos:

```
~$ vim operadores.lua>  
  
a = 4 > 3  
print (a);
```

Salve o arquivo e rode usando o comando já utilizado em exemplos anteriores:

```
~$ lua  
<nome_do_arquivo.lua>
```

Observe que neste caso o programa retorna o valor 1

na tela. Mas caso queira testar resultados diferentes, altere os valores e os operadores para ver como é na prática.

### Operadores lógicos

Muito utilizados na avaliação de expressões booleanas. São eles: and (e), or (ou), not (negação), porém ainda existe o operador “..” que concatena strings. Adeus strcat!!

Vejamos:

```
~$ vim  
operadores_logicos.lua
```

```
a = "Teste de string"  
print (a);  
b = "Esse é um teste"  
print (b);  
c = a..b  
print (c);
```

Agora:

```
~$ lua  
operadores_logicos.lua
```

### Estruturas de controle de fluxo

Nesse aspecto, Lua se parece muito com as demais linguagens. Vejamos:

### Usando o IF:

Como de costume, o if tem diferentes formas de se apresentar. Ele pode se apresentar de maneira bem simples, cuja estrutura se mostra dessa forma:

```
if expressao then  
    bloco  
end
```

Ele também pode aparecer dessa forma:

```
if expressao then
    bloco
else
    bloco2
end
```

E por último temos os chamados ifs aninhados:

```
if expressao then
    bloco
else expressao 2 then
    bloco2
else
    bloco_default
end
```

### Laços de controle com tomada de decisão

Os laços de controle são usadas para que intruções sejam repetidas ou não, de acordo com o retorno do teste de uma determinada expressão. Cada repetição de um laço é chamada de iteração. O teste de uma expressão pode retornar tanto nil, que torna a expressão negativa, quanto qualquer outro valor tornando-a assim verdadeira. Geralmente quando o teste de uma expressão retorna nil, o laço é quebrado, ou

seja, o programa salta para a próxima instrução depois do laço. Vejamos os laços disponíveis em Lua:

#### While

O while testa uma expressão e executa o bloco de comandos caso a mesma seja verdadeira. Sua sintaxe é a seguinte:

```
while expressao do
    bloco
end
```

#### Repeat

Temos também o laço repeat , que ao invés de testar a expressão logo no início, executa todo o bloco pelo menos uma vez, e só então testa a expressão. Caso a expressão seja verdadeira, ocorrerá mais uma iteração. Sua sintaxe é a seguinte:

```
repeat
    bloco
until expressao
```

#### Funções

A linguagem Lua trata as funções como valores, ou seja, podem ser armazenadas em variáveis, passadas como parâme-

tos para funções e ainda servirem de retorno para outras funções.

As funções podem ser declaradas da seguinte forma:

```
function nome ([lista-
    de-parâmetros])
    bloco
end
```

Posteriormente abordarei aqui mais detalhes sobre o uso das funções e também a API Lua.

Por enquanto é só isso pessoal, nos vemos nos próximos capítulos de nossa aventura. E não deixem de acompanhar-nos, nesse mesmo horário e nesse mesmo canal. Até a próxima! 



LÁZARO REINA é usuário Linux, estudante C/C++, Lua, CSS, PHP. Integrante do EESL, ministra palestras e mini-cursos em diversos eventos de Software Livre.

## Java™ Mobile, Media & eMbedded Developer Days

# M3DD Latin America

A chamada de trabalhos já está aberta!  
Envie sua proposta para  
[contato@m3ddla.com.br](mailto:contato@m3ddla.com.br)

Evento latino-americano de tecnologias móveis e embarcadas

El evento latinoamericano de tecnologías móviles y embedded

Mais informações:  
<http://www.m3ddla.com.br>

# AEROTARGET

## Parte 1

Por David A. Ferreira

Você sempre teve vontade de fazer seus próprios jogos? E nunca soube como?

Pois bem, chegou sua vez, darei inicio agora a uma série de artigos sobre Desenvolvimento de Jogos com Software Livre. Em nosso artigo(tutorial), usaremos apenas ferramentas livres, as quais podem ser utilizadas tanto em GNU/Linux quanto no Ms-Windows. Para isto neste primeiro artigo, criaremos um protótipo de um jogo de avião, bastante simples o qual servirá de base para as próximas edições.

Para iniciar, você deve ter instalado em seu computador os seguintes software:

CodeBlocks, uma IDE multiplataforma, a qual pode ser encontrado nos repositórios das principais distribuições ou no site <http://www.codeblocks.org>

SDL, a biblioteca multimídia utilizada pelas principais aplicações multimídia no Linux, e em especial nos jogos. Geral-

mente a SDL vem instalada por padrão em todas as distribuições GNU/Linux, ou então no site <http://www.libsdl.org>

GBFramework, um framework nacional criado para auxiliar no desenvolvimento de jogos 2D, com foco na orientação a objetos e no suporte a multiplataforma. O qual deve ser obtido no site <http://pj-moo.sourceforge.net>

Feita a instalação e Configuração dos aplicativos acima, iremos agora realizar o download do arquivo “aerotarget\_cap01.zip”, o qual está disponível no seguinte endereço:

<http://pj-moo-aerotarget.googlecode.com>

Descompacte o arquivo “aerotarget\_cap01.zip”, e observe a figura 1.

Será criada uma estrutura de diretório, onde no diretório principal, você irá encontrar o arquivo “AeroTarget.cbp”, o qual deverá ser aberto pelo CodeBlocks.

O GBFramework, cria uma estrutura de diretório para auxiliar na organização do seu jogo onde, temos:

- src, arquivos de código fonte

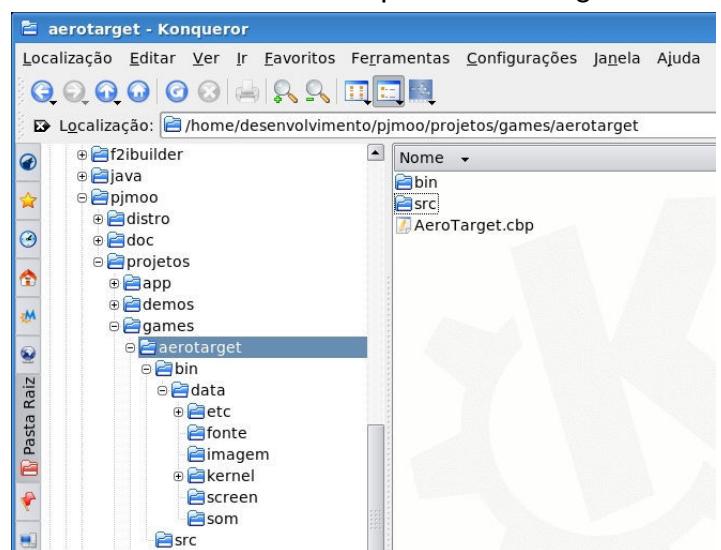


Figura 1: Estrutura de Diretórios

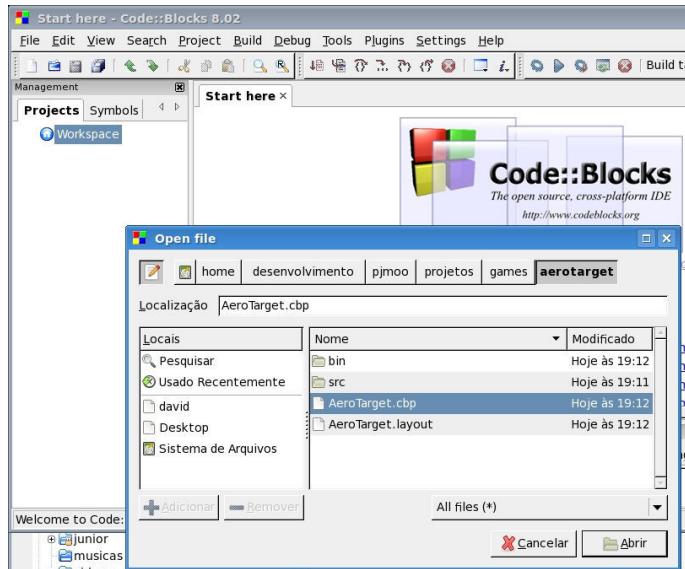


Figura 2: Carregando Projeto

(.cpp e .h);

- bin, local onde será gerado o arquivo binário para execução;
- etc, local sugerido para arquivos de configuração;
- fonte, local que contém os arquivos de fontes(letras);
- imagem, local dos sprites e backgrounds;
- screen, local onde é salvo automaticamente as capturas de telas do jogo;
- som, local dos arquivos de música e efeitos sonoros;
- kernel, diretório especial no qual está localizado os arquivos de recursos (imagens, fontes) requeridos pelo framework.

Agora abra o Code::Blocks, e procure pelo arquivo “AeroTarget.cbp”, conforme mostrado na figura 2.

Ao carregar o projeto, veremos que o mesmo já possui o arquivo “cap\_01.cpp”, o qual contém uma estrutura mínima para uma aplicação GBF, inclu-

sive é possível compilar e testar apenas com este código inicial. Lembre-se de selecionar em Build Target o sistema operacional que você utilizará para compilar o código.

Antes de iniciarmos

o desenvolvimento do nosso jogo, devemos observar o código base do arquivo “cap\_01.cpp”, conforme o código 01.

Onde temos:

- linha 5: função main, utilizada em aplicações c e c++;
- linha 7: declaração do objeto do tipo do GBFramework;
- linha 9: configuração do caminho(path) da aplicação;
- linha 10: título da janela do jogo;
- linha 11: inicialização do jogo, com a configuração para tela de 640x480, com suporte a 16 bits de cores, em modo janela, utilizando o controlador de ciclos no valor de 30 frames por segundo;
- linha 12: configura como será utilizado o teclado e mouse, se será exclusivo para o jogo ou compartilhado com outras aplicações;
- linha 15: carrega o arquivo de imagens para o gerenciador do framework, criando uma ima-

gem com o apelido de “sprites”. O caminho das imagens é relativo, já que na linha 9, informamos para o framework o caminho de sua execução; linha 17: criamos um laço para manter nosso jogo em execução infinita; linha 18 a 20: criamos o controle para permitir encerrar nosso jogo ao pressionarmos a tecla “ESC”; linha 21: utiliza uma fonte padrão para escrevermos o texto em tela, na posição de 10 pixels na horizontal, por 460 pixels na vertical; linha 23: atualiza à tela, ou seja, redesenha as imagens; linha 24: limpa a tela;

Podemos observar como é simples criarmos, um esqueleto para nosso jogo com o GBFramework. Para verificar, se seu ambiente de desenvolvimento está completo, você pode compilar e executar este primeiro fragmento de código. Se tudo ocorrer bem, irá aparecer uma janela, para encerrar basta pressionar a tecla “ESC”.

Iremos agora iniciar o nosso jogo propriamente dito, para isso criaremos a classe Jogador, você pode ir no assistente “Class wizard”, localizado no menu Plugins, conforme pode ser visto na figura 3.

Na janela do assistente, conforme pode ser visto na figura 4, entre com as seguintes informações:

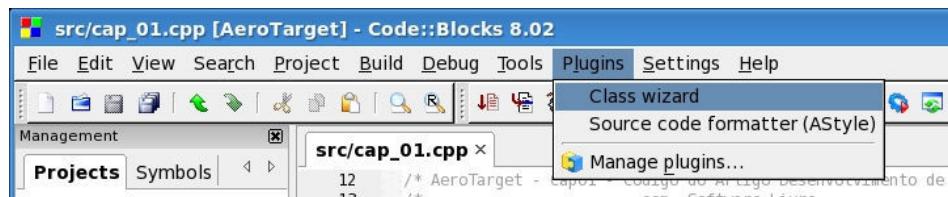


Figura 3: Class wizard

Class name: Jogador  
 Has destructor  
 Virtual destructor  
 Header and implementation file shall be in same folder  
 Folder: /caminho/projeto/aerotarget/src

As outras informações são preenchidas automaticamente, basta agora você clicar no botão “Create”.

Substitua, o código gerado nos arquivos Jogador.h e Jogador.cpp, pelos fragmentos de código correspondente.

Agora repetindo os passos utilizados para a criação da classe pelo “Class wizard”, crie a classe Inimigo, e logo após substitua o conteúdo dos arquivos Inimigo.h e Inimigo.cpp , pelos fragmentos correspondentes:

Explicando o código utilizado nas classes Jogador e Inimigos, temos:

Ambas as classes herdam da classe Personagem (Personagem::Personagem), e com isso ganham vários comportamentos e estados;

Por serem subclasses de Personagem, precisam implementar o método “acao”, o

qual é utilizado para executar o código de ação do personagem, como por exemplo o movimento;

Para associar as imagens aos respectivos personagens, devemos recuperar do gerenciador do framework, uma “Sprite-Factory”, a qual recebe como parâmetro, o apelido que definidos para guardar os sprites;

Cada personagem, possui uma animação principal, a qual pode ser configurada por meio do método “adicionarSpritePrincipal”;

Para concluirmos nosso primeiro exemplo, iremos agora adicionar ao jogo os inimigos e o jogador, efetue as seguintes modificações:

1.Antes do método main, adicione o  
`#include <Inimigo.h> e <Jogador.h>, além do #define TOTAL_INIMIGO,` conforme o código 6.

2.Adicione após o carregamento da imagem, uma variável

para representar o jogador e um vetor para representar os inimigos, não esquecendo de instanciar os objetos. Conforme pode ser visto no código 7.

3.No bloco while, adicione o controle de ações e desenhos dos inimigos e do jogador. Podemos observar que cada um contém dois métodos, os quais devem ser agrupados de acordo com o tipo, ou seja, primeiro são chamados os métodos de atualização dos objetos, e somente depois os métodos de desenho, conforme visto no código 8.

4.Para encerrar, não esqueça de destruir os objetos que você instanciou, para isso após o bloco while, adicione a chamada a delete, conforme o código 9.

Realizada estas modificações, basta compilar e executar o projeto, onde deverá aparecer uma janela, com o nosso jogo sendo executado,

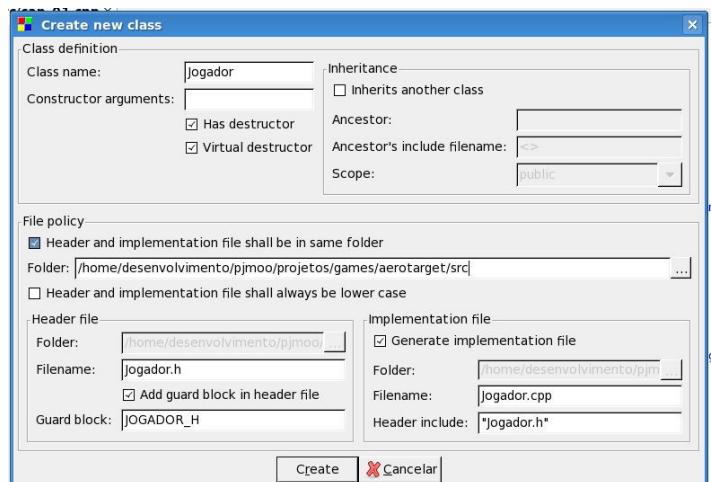


Figura 4: Assistente para Criação de Classe



Figura 5: Tela do Jogo

como na figura 5.

Com isso, encerro o nosso primeiro artigo da série sobre Desenvolvimento de Jogos com Software Livre, espero que com este artigo, você possa dar seus primeiros passos, nesta vasta jornada rumo ao mundo do desenvolvimento de

jogos. Enquanto você espera a próxima edição, tente praticar um pouco, e com base no que foi apresentando, tente inserir no jogo as características básicas que estão faltando, como por exemplo: tiros, colisões, energia, mini chefes e etc. Será um bom exercício de auto desenvolvimento, e permitirá que você possa comparar sua implementação com a da próxima edição. 

## Códigos utilizados neste artigo:

```
1.*****  
2./* AeroTarget - Cap01 - Código do Artigo Desenvolvimento de Jogos  
 */  
3./* com Software Livre */  
4.*****  
5.int main(int argc, char* argv[]){  
6{  
7.    GBF::GBFramework frameworkGBF;  
8.  
9.    frameworkGBF.setPath(argv[0]);  
10.   frameworkGBF.setTitulo("AeroTarget - Cap 01","GBFramework &  
Revista Espírito Livre");  
11.  
12.   frameworkGBF.iniciar(640,480,16,false,GBF::Kernel::FPS::FPS_LIMIT  
ADO);  
13.   frameworkGBF.inputSystemCore->setControleExclusivo(SDL_GRAB_OFF);  
14.//carregando imagens  
15. frameworkGBF.graphicSystemCore->graphicSystem-  
>imageBufferManager-> carregar("sprites","//data//imagem//sprites.png");  
16.  
17.     while (true){  
18.         if (frameworkGBF.inputSystemCore->inputSystem-  
>teclado->isKey(SDLK_ESCAPE)){  
19.             break;  
20.         }  
21.     frameworkGBF.writeSystem->  
escrever(GBF::Kernel::Write::WriteManager::defaultFont,10,460,"AeroTar  
get - Cap01 : GBFramework & Revista Espírito Livre");  
22.         //realiza refresh, fps, flip  
23.         frameworkGBF.atualizar();  
24.         frameworkGBF.graphicSystemCore->clear();  
25.     }  
26.     return 0;  
27.}
```

**Código 1: cap\_01.cpp**

```
28.#ifndef JOGADOR_H  
29.#define JOGADOR_H  
30.  
31.#include <GBF/Personagem.h>  
32.#include <GBF/SpriteFactory.h>  
33.#include <GBF/InputSystem.h>  
34.  
35.class Jogador : public Personagem::Personagem  
36{  
37.    public:  
38.        Jogador();  
39.        virtual ~Jogador();  
40.        void acao(GBF::Kernel::Input::InputSystem * input);  
41.    protected:  
42.        private:  
43.};  
44.#endif // JOGADOR_H
```

**Código 2: Jogador.h**

```
45.#include "Jogador.h"  
46.  
47.Jogador::Jogador()  
48.{  
49.    GBF::Imagem::SpriteFactory *spriteFactory = new  
GBF::Imagem::SpriteFactory("sprites");  
50.    adicionarSpritePrincipal(spriteFactory-  
>criarSpritePersonagem(0,0,59,43,3,6));  
51.    getSpritePrincipal()->setQtdDirecoes(1);  
52.    getSpritePrincipal()->animacao.setAutomatico(true);  
53.    delete (spriteFactory);  
54.  
55.    setPosicao(320,400);  
56.}  
57.Jogador::~Jogador()  
58.{  
59.}  
60.void Jogador::acao(GBF::Kernel::Input::InputSystem * input)  
61.{  
62.    if (input->teclado->isKey(SDLK_RIGHT)){  
63.        posicao.x+=4;  
64.    } else if (input->teclado->isKey(SDLK_LEFT)){  
65.        posicao.x-=4;  
66.    }  
67.}
```

**Código : Jogador.cpp**

```

68.#ifndef INIMIGO_H
69.#define INIMIGO_H
70.
71.#include <GBF/Personagem.h>
72.#include <GBF/SpriteFactory.h>
73.#include <GBF/InputSystem.h>
74.
75.class Inimigo : public Personagem::Personagem
76.{}
77.    public:
78.        Inimigo();
79.        virtual ~Inimigo();
80.        void acao(GBF::Kernel::Input::InputSystem * input);
81.    protected:
82.    private:
83.        void iniciar();
84.        int velocidade;
85.};
86.#endif // INIMIGO_H

```

**Codigo 4: Inimigo.h**

```

87.#include "Inimigo.h"
88.
89.Inimigo::Inimigo()
90.{
91.    GBF::Imagem::SpriteFactory *spriteFactory = new
GBF::Imagem::SpriteFactory("sprites");
92.    adicionarSpritePrincipal(spriteFactory-
>criarSpritePersonagem(0,77,32,32,3,6));
93.    getSpritePrincipal()->setQtdDirecoes(1);
94.    getSpritePrincipal()->animacao.setAutomatico(true);
95.    delete (spriteFactory);
96.    iniciar();
97.}
98.Inimigo::~Inimigo()
99.{
100.}
101.void Inimigo::acao(GBF::Kernel::Input::InputSystem * input)
102.{
103.    posicao.y+=velocidade;
104.    if (posicao.y>640){
105.        iniciar();
106.    }
107.}
108.void Inimigo::iniciar()
109.{
110.    GBF::Dimensao d = getDimensao();
111.    GBF::Ponto p;
112.    p.y= 0 - d.h;
113.    p.x= rand() % (640 - d.w);
114.
115.    setPosicao(p);
116.    velocidade=2+rand()%5;
117.}

```

**Codigo 5: Inimigo.cpp**

```

118.#include <vector>
119.#include <iostream>
120.
121.#include "Inimigo.h"
122.#include "Jogador.h"
123.#define TOTAL_INIMIGO 3

```

**Codigo 6: Include e Define**

```

124.//carregando imagens
125.    frameworkGBF.graphicSystemCore->graphicSystem-
>imageBufferManager->carregar("sprites","/data/imagem//sprites.png");
126.
127.    Jogador * aviao = new Jogador();
128.    Inimigo * inimigo[TOTAL_INIMIGO];
129.
130.    for (int i=0; i<TOTAL_INIMIGO; i++){
131.        inimigo[i] = new Inimigo();
132.    }

```

**Codigo 7: Declarando e inicializando**

```

133.    while (true) {
134.        if (frameworkGBF.inputSystemCore->inputSystem-
>teclado->isKey(SDLK_ESCAPE)){
135.            break;
136.        }
137.
138.        for (int i=0; i<TOTAL_INIMIGO; i++){
139.            inimigo[i]->acao(NULL);
140.        }
141.
142.        aviao->acao(frameworkGBF.inputSystemCore->inputSystem);
143.
144.        for (int i=0; i<TOTAL_INIMIGO; i++){
145.            inimigo[i]->desenhar();
146.        }
147.
148.        aviao->desenhar();

```

**Codigo 8: Processando**

```

149.        //realiza refresh, fps, flip
150.        frameworkGBF.atualizar();
151.        frameworkGBF.graphicSystemCore->clear();
152.    }
153.
154.    delete(aviao);
155.    for (int i=0; i<TOTAL_INIMIGO; i++){
156.        delete(inimigo[i]);
157.    }

```

**Codigo 9: Finalizando**

## Maiores informações:

Download aerotarget\_cap01.zip:

<http://pjmoor-aerotarget.googlecode.com>

Site PJMOO:

<http://pjmoor.sourceforge.net>

Site CodeBlocks:

<http://www.codeblocks.org>



**DAVID ALMEIDA FERREIRA** é Especialista em Engenharia de Software com Ênfase em Padrões de Software(UECE), É Arquiteto de Software do Banco do Nordeste do Brasil (BNB). É integrante do Projeto de Software Livre Ceará(PSL-CE) e do Grupo de Desenvolvedores de Jogos do Ceará(GDJCE).

# Scrum: uma aproximação ágil ao desenvolvimento de software

Por Marcello Duarte

Flávio Takemoto - sxc.hu

Desenvolver software é um exercício complexo. Não apenas porque envolve quase sempre tecnologias que estão em constante renovação, mas especialmente pelo seu carácter multi-disciplinar.

Desenvolver software é um exercício complexo. Não apenas porque envolve quase sempre tecnologias que estão em constante renovação, mas especialmente pelo seu carácter multi-disciplinar. Pelo menos duas dimensões, técnica e tática, têm que interagir. As regras do negócio têm que ser comunicadas claramente e traduzidas em código por quem desenvolve. E como se não bastasse essas regras mudam, as necessidades mudam, o mundo muda a um passo cada vez mais frenético. A capacidade de responder rapidamente não é um valor diferencial, é um factor crítico de sucesso.

## Scrum e Agile

Durante as décadas de 80 e 90 muitas metodologias foram aparecendo como uma resposta natural à esta realidade, algumas inspiradas pelo Toyota way e pelos processos de manufatura Lean. A partir da publicação do Agile Manifesto em 2001, algumas dessas metodologias passaram a estar directamente associadas ao movimento Agile. Agile é portanto uma referência à princípios e directrizes relacionadas com o desenvolvimento. Scrum é uma framework para gestão de projectos que procura seguir esses princípios.

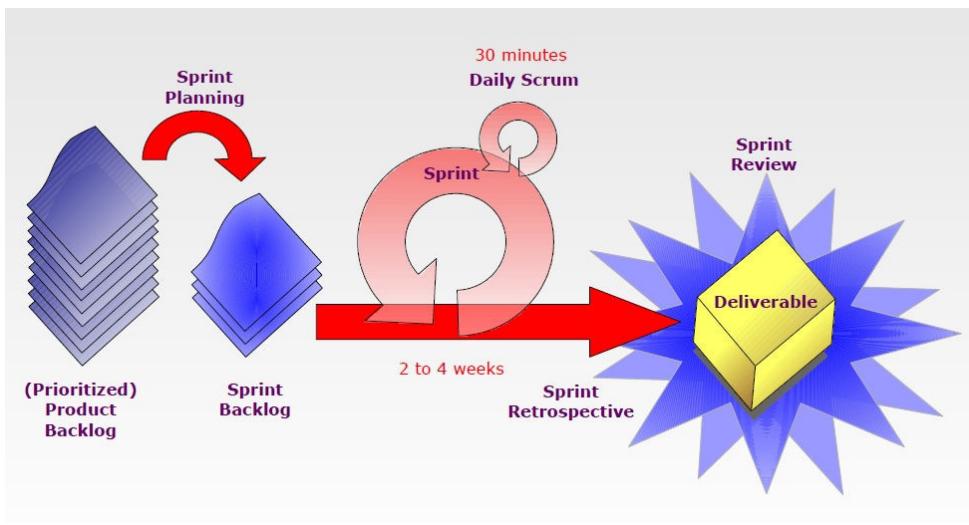


Figura 1 - O processo

## O processo em poucas linhas

Em suma, um projecto Scrum pode ser descrito como segue. Uma necessidade de negócio ou visão é traduzida numa lista de requisitos (Product Backlog). Esta lista não deve ser exaustiva nem detalhada. Os items são priorizados pelo Product Owner (Proprietário do Produto) e estimados por uma equipa entre 4 e 9 membros (Team Members). O projecto é dividido em Sprints (sub-projectos), cuja duração pode variar de 1 a 4 semanas. No começo de cada Sprint o Product Owner define o objectivo daquela iteração e a Equipa compromete-se a desenvolver um entregável perfeitamente funcional no fim do período. Não se trata de um protótipo, nem prova de conceito, mas de uma fracção perfeitamente funcional do software. Cada Sprint terá um Sprint Backlog com a sua sub-lista de requisitos, cuja estimativa coincide com a duração estabelecida para cada Sprint.

## O Scrum Master

Scrum é uma framework leve com uma hierarquia de verticalidade mínima, apenas 3 papéis: Product Owner, Team Member e Scrum Master.

O Scrum Master é o pivo do Scrum. Não devemos confundi-lo com o gestor de projectos tradicional. Em Scrum os 3 papéis são gestores. A equipa é responsável e compromete-se com os prazos e proactivamente toma decisões a respeito de quem e como as tarefas são executadas.

O Scrum Master funciona como um facilitador no processo. Suas responsabilidades são: difundir as regras do Scrum, assegurar visibilidade do projecto, assistir a equipa sempre que um impedimento surge, servir de escudo entre o Product Owner e a equipa, assegurar a qualidade, produzir os artefactos do Scrum, realizar as cerimónias do Scrum.

## Artefactos

Um artefacto em Agile é qualquer sub-produto que deriva das actividades e processos levados a cabo durante um projecto mas que não é necessariamente parte do produto. Por exemplo o gráfico Burndown, o quadro de tarefas e o Product Backlog. É tarefa do Scrum Master assegurar que os artefactos sejam distribuídos por todos os stakeholders para assegurar a máxima visibilidade do projecto por todos os interessados

## Cerimónias

Além da visibilidade outros dois conceitos importantes em Scrum é inspecção e adaptação. O Scrum Master encarrega-se de promover as seguintes Cerimónias: Scrum Planning, Daily Scrum, Scrum Review e Scrum Retrospective. Estas cerimónias são as reuniões de projecto do Scrum. Cada uma com um propósito bem definido. Planeamento, sincronização diária, inspecção e adaptação. Além de um objectivo definido cada cerimónia também tem um universo de tempo muito bem definido. Esse conceito chama-se time box.

O Scrum Planning é composto de duas fases. Cada uma dessas fases tem um time box de 4 horas (podendo variar dependendo do projecto). Na primeira o Product Owner explica cada um dos items no

Product Backlog e responde a qualquer questão que possa surgir para a equipa. Na segunda a equipa parte os itens (User Stories) em tarefas e estima o tempo necessário para executar cada uma delas, atribuindo um responsável por cada tarefa. No fim o Product Owner volta a validar os itens seleccionados segundo os objectivos do Sprint.

O Daily Scrum é uma oportunidade que a equipa tem para sincronizar. Tem normalmente um time box de 15 ou 30 minutos. Durante esta cerimónia cada um dos membros responder a três questões: Que fizeste deste a última sincronização? No que vais trabalhar até a próxima sincronização? Que impedimentos o previne de concluir a tarefa? Outro propósito que se cumpre nestas sessões é que o Scrum Master como facilitador, toma nota e responsabiliza-se por apoiar o membro da equipa na resolução do impedimento.

É no Sprint Review que o Product Owner tem a oportunidade de inspecionar o resultado da iteração. Durante esta sessão a equipa demonstra o software entregável. O Sprint Review pode direcionar o Pro-

duct Owner na sua decisão para a visão do próximo Sprint.

Sprint Retrospective é onde adaptação entra no processo. Cada membro da equipa, o Scrum Master e opcionalmente o Product Owner pode dizer o que correu bem e o que pode melhorar. A informação recolhida é transformada em activo do projecto (lessons applied).

Embora tenha apresentado Scrum e Agile dentro do contexto de desenvolvimento de software, simplesmente devido a minha área de especialidade estar ligado com esta indústria, Scrum pode ser aplicado a virtualmente qualquer indústria. Tenho encontrado pessoas que aplicam Scrum, por exemplo, em equipas que trabalham com projectos legislativos, manuais de contabilidade, e muitas outras áreas.

Scrum e Agile são especialmente relevantes em ambientes de mudança, ou quando uma parte do produto pode ser posta em produção antes da versão final estar terminada e proporcionar assim uma diminuição do tempo necessário para o retorno do investimento. Outra aplicação é quando uma parte da aplicação possui um certo risco e pode ser priorizada ante as demais, permitindo

decisões estratégicas para a direcção ou vida do produto. 

### Referências

Schwaber, Ken (1 de Fevereiro de 2004). Agile Project Management with Scrum

### Maiores informações:

#### Site Control Chaos:

<http://www.controlchaos.com>

#### Site Scrum Alliance:

<http://www.scrumalliance.org>



**MARCELLO DUARTE** é o autor da premiada Quinoa framework. Licenciou-se pela Universidade Autónoma de Lisboa em Informática de Gestão, tendo se especializado em Gestão de Projectos (PMP e Scrum Master Certified). Actualmente trabalha como Engenheiro de Software para a Ibuildings UK, em Londres.

Em Salvador - Bahia,  
Dois Grandes Eventos  
29 e 30 de Maio.  
[Entrada Gráuita]



**III Encontro Nordestino de Software Livre  
IV Festival de Software Livre da Bahia**

# E você? Já escolheu o seu controlador de versões?

Por Evaldo Junior

Calma não precisa escolher um só, as opções estão aí, use e abuse!

Majoros Attila - sxc.hu

É até engraçado ver como começamos na vida do desenvolvimento de software sempre usando o estilo “vou guardar isso em um diretório no meu computador”, e lá se vão alguns diretórios para guardar códigos de diferentes projetos, o que não é necessariamente uma coisa ruim, quando você trabalha sozinho ou tem projetos realmente pequenos.

Mas quando alguém resolve contribuir com o seu projeto a coisa começa a complicar, imagine a cena, um amigo lhe pede o código, você compacta e envia tudo por e-mail, esse

amigo faz as alterações que julgar necessárias e implementa novas funcionalidades, então ele lhe envia tudo de novo, você vê as diferenças e as aplica ao código principal... Chato isso, não? Mas seria ainda mais chato se fosse uma equipe de, digamos, apenas três pessoas. E o que aconteceria então se a equipe crescesse?

Toda essa história de equipe já assusta um pouco, mas também existe o problema das versões, como voltar a ter o sistema de, digamos, um ano atrás? Ou mesmo desfazer uma grande parte de código que você estava trabalhando?

Se você já passou por esses problemas sabe como é complicado, se ainda não passou é melhor começar a se prevenir e já adotar um sistema de controle de versões. Mas agora vem a pergunta, o que é um sistema de controle de versões mesmo?

Os sistemas controladores de versão[1] (version control system) são utilizados para controlar as várias versões principalmente de software, mas podem ser usados para controlar versões de vários tipos de documentos digitais.

Existem controladores de versão livres e também os pro-

prietários, eu, particularmente, já trabalhei com o Subversion[2] e com o Git[3], mas posso indicar opções como o famoso CVS[4] e o Mercurial[5]. E o legal é que não ficamos apenas nisso, existem muitos outros controladores de versão por aí, basta pesquisar, comparar e escolher!

Depois de escolher o seu controlador de versões você poderia muito bem usa-lo apenas localmente, isso lhe daria a oportunidade de navegar entre as versões do seu software, o que pode lhe ajudar muito em várias situações.

Mas vamos falar de colaboração! Qual é a graça de manter seu código só para você? Ainda mais quando falamos de Software Livre?

Existem várias ferramentas para tornar seu código público e ainda utilizar uma solução em controle de versões. Uma ferramente muito conhecida é o Sourceforge[6], nele podemos usar, por exemplo, o CVS, Subverion e Git, além de termos à nossa disposição ferramentas como Wiki e Bug Tracker.

Funcionalidades parecidas podemos encontrar no Google Code[7], onde eu mantendo o GeSpeak[8]. No Google Code contamos com o Subversion como controlador de versão, Wiki, Bug Tracker e contador de downloads.

Um sistema legal que surgiu nos últimos tempos e está

dando o que falar é o GitHub[9], nele podemos trabalhar com o Git, que, aliás, já nasceu famoso, já que é uma criação de Linus Torvalds, conhece? O GitHub não oferece um Bug Tracker, mas a Wiki está lá.

Uma outra ferramenta que posso indicar é o BitBucket[10], nele podemos usar o Mercurial como controlador de versões, além de um Bug Tracker e Wiki.

O interessante dessas ferramentas é que, além de hospedar projetos e oferecer recursos para controlar versões, elas também acabam se tornando redes sociais para desenvolvedores, e é de código em código que o pessoal vai se conhecendo e evoluindo a cada dia.

Ultimamente eu estou bem empolgado com o GitHub e até já o utilizei para desenvolver um projeto pessoal, o WPMLC[11].

Eu recomendo que você conheça pelo menos dois controladores de versão, quanto mais melhor, é claro, mas, por enquanto, eu estou satisfeito com o Subversion e o Git.

Então é isso, escolha o seu controlador de versões, use uma das ferramentas de colaboração para tornar seu código público e boa sorte com seus projetos! Nos encontramos nos repositórios por aí, até a próxima! 

## Maiores informações:

### [1] Artigo na Wikipédia:

[http://pt.wikipedia.org/wiki/Controle\\_de\\_versions](http://pt.wikipedia.org/wiki/Controle_de_versions)

### [2] Site Subversion:

<http://subversion.tigris.org/>

### [3] Site Git:

<http://git-scm.com/>

### [4] Site CVS:

<http://www.nongnu.org/cvs/>

### [5] Site Mercurial:

<http://www.selenic.com/mercurial/wiki/>

### [6] Site Sourceforge:

<http://sourceforge.net/>

### [7] Site Google Code:

<http://code.google.com/>

### [8] Site Projeto GeSpeak:

<http://gespeak.googlecode.com/>

### [9] Site Github:

<https://github.com/>

### [10] Site CVS:

<http://bitbucket.org/>

### [11] Site Projeto WPMLC:

<http://github.com/InFog/wpmultilanguagechanger>



**EVALDO JUNIOR** (InFog) é formado pela Fatec em processamento de dados e atualmente é desenvolvedor, administrador de sistemas e membro da comunidade de software livre.



# IFES Cariacica: Gestão de incidentes de TI com o Ocomon

Por Célio P. Maioli

Slavomir Ulicny - sxc.hu

A unidade de Cariacica (região metropolitana de Vitória) do CEFET-ES[1], atualmente nomeado campus Cariacica do Instituto Federal do Espírito Santo, iniciou suas atividades em agosto de 2006 com o curso técnico pós-médio em ferrovias e uma infraestrutura que precisava de algumas melhorias, inclusive na área de tecnologia da informação. Neste sentido, para prestar serviços educacionais reconhecidamente de qualidade para o público capixaba, o Instituto conta em seu corpo funcional com um grupo de técnicos de nível médio e superior em cada uma de suas nove unidades espalhadas pelo estado.

Com a implantação da unidade de Cariacica, e minha

aprovação no concurso para o cargo de Analista de TI, fui convidado a assumir a coordenação dos trabalhos de implantação da infraestrutura de LAN, necessária para interligar o campus a internet. Quando nossa estrutura estava minimamente funcional - e com o crescimento das demandas de serviços - percebemos que havia a necessidade da implantação de algum sistema capaz de organizar os incidentes de TI da escola. As demandas chegavam por telefone ou email, quando não eram solucionadas por um dos técnicos da equipe que estava de passagem, próximo ao problema. Chamávamos isto de "solução de corredor", já que todos os problemas, sempre urgentes, eram resolvidos sem uma in-



Figura 1: Unidade IFES Cariacica - Fonte: [www.ifes.edu.br](http://www.ifes.edu.br)

terface única entre os setores da escola e a tecnologia da informação.

Portanto, era muito importante começar a padronizar e documentar os procedimentos de atendimento. Além disto, recebíamos críticas por serviços não realizados e por outros, cujo tempo de resposta e atendimento era excessivamente grande. Também havia casos nos quais a solução não cheava sequer ao contento do usuário atendido.

Assim, era urgente implementar alguma ferramenta de controle, mas alguns requisitos precisavam ser preenchidos, dentre eles, a apresentação do sistema, que deveria ser do tipo "web para fácil acesso a todos os usuários"; módulos de relatório; integração com nosso serviço de diretório; ser um

sistema de código aberto para que, se fossem necessários, ocorressem os ajustes mínimos, dada a nossa pequena equipe.

Entre pesquisas em fóruns e sites especializados em software livre, encontramos o Ocomon[2], um sistema de registro de incidentes bastante aderente ao ITIL (referência mundial no gerenciamento de serviços de TI), que já vem sendo usado por várias organizações no Brasil. Baixamos a versão 1.4, uma anterior que a mais recente, pois assim teríamos um sistema mais estável.

O projeto de implementação foi dividido em 5 fases: estrutura básica, adequação do sistema, cadastros, testes de uso e lançamento.

### Fases do Projeto

Iniciamos a fase 1 no início de 2007 com a aquisição de uma máquina que atendesse aos requisitos mínimos para atuar como servidor web. Conseguimos um HP 5150, Athlon 3600+ com 1Gb de RAM e 80GB de HD, ou seja, um desktop que transformamos em servidor. Esta máquina foi suficiente por que não havia interesse no acesso externo ao sistema, ou seja, apenas o público interno a escola poderia registrar problemas de TI. Escolhemos a plataforma LAMP (Linux, Apache, MySql, PHP) para abrigar o sistema, pois seria mais simples a implementação destes e de outros projetos de software livre.

A fase dois correu quase paralelamente à fase um, pois tratava de pequenos ajustes gráficos e funcionais ao site. Inserimos o logo do CEFETES e retiramos alguns campos não interessantes ao nosso contexto, como a etiqueta do produto. Entretanto, a principal alteração no código foi a integração com nosso serviço de diretórios. O sistema CEFETES adota o Active Directory (AD) da Microsoft para gerenciar usuários, grupos e políticas de segurança, por isso era fundamental ao funcionamento do sistema que os usuários pudessem "logar" com mesmo usuário e senha, que já possuíam, para ingressar na rede e no email. Assim que a integração com o AD e demais ajustes fo-

ram feitos, passamos ao cadastramento dos principais problemas e das SLAs (service level agreement), baseando-se em nossa experiência anterior - de solução de incidentes. Dividimos os problemas em três grandes categorias: hardware, software e outros. Em cada categoria consta uma lista de problemas genéricos tais como: Mouse não funciona, sistema operacional não inicia, ou criação de conta de usuário, respectivamente. Fizemos os testes necessários antes de liberar os testes para alguns usuários-chave.

Na fase de testes, incluímos alguns problemas na relação que não constavam originalmente e fizemos alguns pequenos ajustes.

Com os usuários-chave treinados e o apoio da direção, lançamos efetivamente nosso sistema de controle de incidentes de TI em março de 2008. Como nossa unidade era a úni-

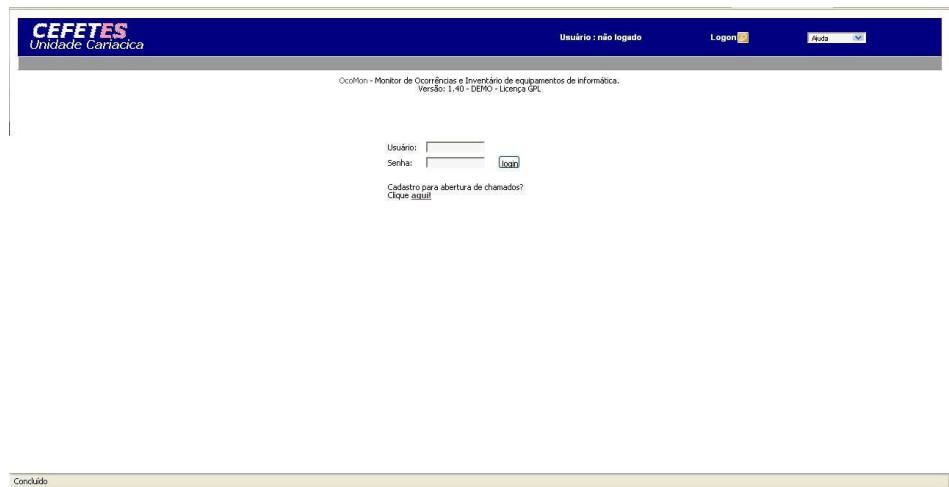


Figura 2: O sistema em funcionamento

ca que oferecia este serviço, houve estranheza e resistência inicial pois os usuários ainda queriam "soluções de corredor", no entanto, com o tempo, conseguimos uma interface única, na qual poderíamos documentar os incidentes, ter uma fila rígida para atendimentos e, no fim de um período, gerar relatórios gerenciais de apoio à decisão adotada. Atualmente outras unidades do sistema estão buscando soluções semelhantes, usando ou não o

Ocomon, mas a unidade de Carriacica saiu na vanguarda e ainda é a única que tem uma base de dados com mais de um ano de incidentes de TI. Portanto, é imprescindível para uma instituição, do tamanho e da importância do Instituto Federal, um serviço de TI pró-ativo, que possa entregar à direção informações gerenciais capazes de nortear ações de capacitação e prevenção a problemas, e sistemas como o ocomon são chave para isto. 

É imprescindível para uma instituição um serviço de TI pró-ativo, que possa entregar a direção informações gerenciais capazes de nortear ações de capacitação e prevenção a problemas...

Célio P. Maioli

**Maiores informações:**  
[1] Site CEFET-ES:  
<http://www.cefetes.br>

[2] Site Oficial Ocomon:  
<http://ocomonphp.sourceforge.net>



**CÉLIO P. MAIOLI** é engenheiro de computação com Mestrado em Engenharia Elétrica pela UFES e atualmente é professor do campus Serra do IFES.

# Primeiras impressões do dotProject

Por Saymon Castro

Um projeto é um empreendimento com características próprias, tendo princípio e fim, conduzidos por pessoas, para atingir metas estabelecidas dentro de parâmetros de prazo, custo e qualidade. Um projeto é um empreendimento temporário, cujo objetivo é criar um produto ou serviço, distinto e único.

A gestão de projetos é aplicação dos conhecimentos, habilidades, ferramentas e técnicas com o objetivo de atingir ou até mesmo exceder as necessidades e expectativas dos clientes e demais partes interessadas do projeto, por este motivo, portanto nenhum projeto poderá ser bem sucedido se não for bem gerenciado.

No mercado, existem softwares como por exemplo MS Project (Microsoft) que são referência quando o assunto é gerenciamento de projetos. Tais softwares auxiliam na concepção, planejamento e controle dos projetos.

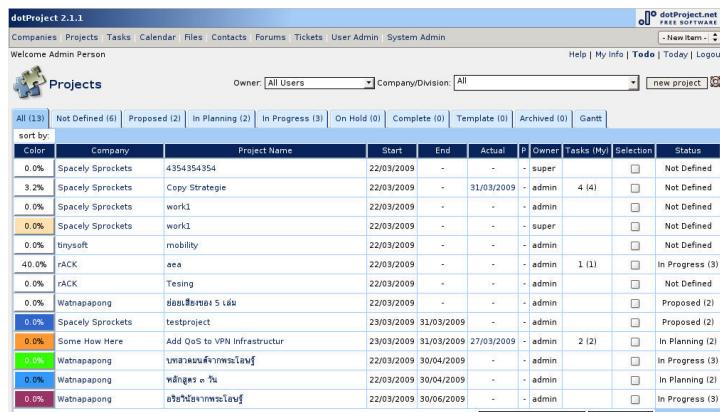
Pesquisando sobre alternativas de software ao Project da Microsoft, descobri o projeto opensource dotProject ([www.dotproject.net](http://www.dotproject.net)). O dotProject é uma ferramenta de gerenciamento de projetos, cuja interface é web (desenvolvido em php e banco de dados MySQL), que facilita a interatividade da equipe com a atualização da execução das tarefas de um projeto.

A instalação é relativamente simples para quem possui certo conhecimento de redes. Bas-

ta ter um servidor web + php e o banco de dados MySQL instalados. Existem os módulos Add-on, que podem ser encontrados no site oficial, tais módulos são desenvolvidos por usuários que servem de extensão para o projeto. Um exemplo de add-on é a tradução do dotProject para o português Brasil.

O primeiro acesso ao dotProject será feito através do usuário: admin, senha: passwd. A primeira tela, após a autenticação é parecida com a figura 1.

Na parte superior, veremos a versão do dotProject (neste caso é 2.1.2), logo abaixo, estarão os links principais de navegação. À direita temos um calendário e mais ao centro a parte mais operacional do sistema.



The screenshot shows the dotProject 2.1.2 dashboard. At the top, there's a navigation bar with links for Companies, Projects, Tasks, Calendar, Files, Contacts, Forums, Tickets, User Admin, and System Admin. Below that is a welcome message for 'Admin Person'. The main area is titled 'Projects' and contains a table with 13 rows. The columns are: All (13), Not Defined (6), Proposed (2), In Planning (2), In Progress (3), On Hold (0), Complete (0), Template (0), Archived (0), Gantt, Selection, and Status. Each row represents a project with details like name, start date, end date, owner, tasks, and status. For example, the first project is 'Spacely Sprockets' with a status of 'Not Defined'.

All (13)	Not Defined (6)	Proposed (2)	In Planning (2)	In Progress (3)	On Hold (0)	Complete (0)	Template (0)	Archived (0)	Gantt	Selection	Status
Color	Company	Project Name	Start	End	Actual	P	Owner	Tasks (My)			
0.0%	Spacely Sprockets	4354354354	22/03/2009	-	-	-	super	4 (4)	<input type="checkbox"/>		Not Defined
3.2%	Spacely Sprockets	Copy Strategie	22/03/2009	-	31/03/2009	-	admin	4 (4)	<input type="checkbox"/>		Not Defined
0.0%	Spacely Sprockets	work1	22/03/2009	-	-	-	admin		<input type="checkbox"/>		Not Defined
0.0%	Spacely Sprockets	work1	22/03/2009	-	-	-	super		<input type="checkbox"/>		Not Defined
0.0%	tinysoft	mobility	22/03/2009	-	-	-	admin		<input type="checkbox"/>		Not Defined
40.0%	rACK	aea	22/03/2009	-	-	-	admin	1 (1)	<input type="checkbox"/>		In Progress (3)
0.0%	rACK	Tesing	22/03/2009	-	-	-	admin		<input type="checkbox"/>		Not Defined
0.0%	Watnapapong	ทดสอบระบบ 5 ล้าน	22/03/2009	-	-	-	admin		<input type="checkbox"/>		Proposed (2)
0.0%	Spacely Sprockets	testproject	29/03/2009	31/03/2009	-	-	admin		<input type="checkbox"/>		Proposed (2)
0.0%	Some How Here	Add QoS to VPN Infrastructur	29/03/2009	31/03/2009	27/03/2009	-	admin	2 (2)	<input type="checkbox"/>		In Planning (2)
0.0%	Watnapapong	ทดสอบระบบค้าขายออนไลน์	22/03/2009	30/04/2009	-	-	admin		<input type="checkbox"/>		In Progress (3)
0.0%	Watnapapong	ทดสอบระบบค้าขายออนไลน์	22/03/2009	30/04/2009	-	-	admin		<input type="checkbox"/>		In Planning (2)
0.0%	Watnapapong	ทดสอบระบบค้าขายออนไลน์	22/03/2009	30/06/2009	-	-	admin		<input type="checkbox"/>		In Progress (3)

Figura 1 - Lista de projetos

## Principais Recursos:

### Gerenciamento de Clientes e Empresas

É possível cadastrar e gerir diversas empresas e clientes, com diversos departamentos, pessoas responsáveis pelos departamentos, equipes entre outros.

Company Name	Active Projects	Archived Projects	Type
FACK	2	0	Supplier
Running S.r.l.	0	0	Not Applicable
Some How Here	1	0	Not Applicable
Spacely Sprockets	5	0	Not Applicable
tinysoft	1	0	Not Applicable
Watnapapong	4	0	Not Applicable

Figura 2 - Gerenciamento de empresas

### Gerenciamento de usuários

É possível configurar diversas permissões dos usuários em relação ao sistema, visualizar os usuários online e offline, em que momento e local (data/hora e IP) eles entraram no sistema e as sessões abertas no momento.

Login Name	Real Name	Company
admin	Person, Admin	
guest	User, Guest	
super	User, Super	

Figura 3 - Gerenciamento de usuários

### Integração com servidor de email

É possível integrar o dotProject a um servidor de email, assim as novas atividades/projetos podem ser avisados via email.

### Lista de Tarefas em formato Hierárquico

As tarefas podem assumir um caráter hie-

Task Name	Task Creator	Assigned Users	Start Date	Duration	Finish Date
JACK :: aea	admin	admin (100%)	-	1 hours	-
adding some support features	admin	admin (100%)	-	1 hours	-
programmerung	admin	admin (100%)	22/03/2009 08:45	1 hours	24/03/2009 17:00

Figura 4 - Lista de tarefas

rárquico, por exemplo, em um projeto X a tarefa 1 é pré-requisito para execução da tarefa 2, assim podendo gerar relatórios do tipo Gantt.

Figura 5 - Relatório do tipo Gantt

### Repositório de Arquivos

É possível anexar arquivos referentes a cada tarefa de um projeto.

File Name	Description	Versions	Category	Folder	Task Name	Owner	Size	Type	Date
DSCF3737.JPG	mmmm	1.00	Document	krystian	Admin Person	1.23 Mb	jpeg		22/03/2009 09:11
Testing.doc		1.01 (2)	Document	Root	Admin Person	8 Kb	octet-stream		22/03/2009 07:13

Figura 6 - Repositório de arquivos

### Fórum

É comum que na concepção, planejamento, controle e desenvolvimento de um projeto hajam diversos debates. Pensando nesse aspecto o dotProject possui um fórum para cada projeto.

Forum Name	Topics	Replies	Last Post Info
My forum	2	1	22/03/2009 09:18 > 999

Figura 7 - Fórum

### Calendário

O calendário do dotProject registra todas os eventos e atividades referentes aquele dia, ajudando muito nessa gestão.

# ADMINISTRAÇÃO · PRIMEIRAS IMPRESSÕES COM O DOTPROJECT

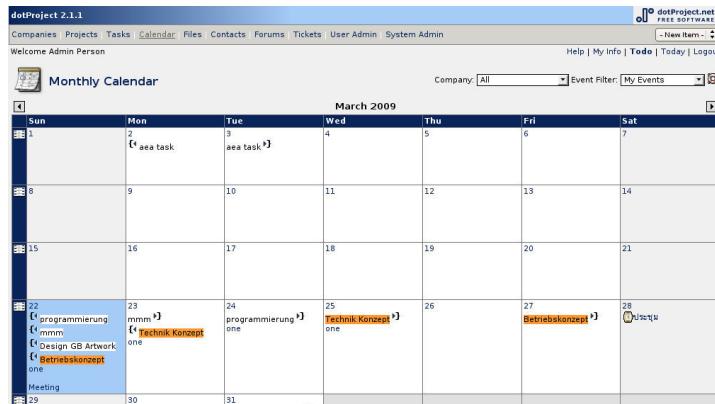


Figura 8 - Calendário

## Sistema de Chamados

É possível criar chamados referentes a uma empresa e projeto. Esse recurso auxiliará a enviar observações, detalhes ou perguntas sobre aquele projeto.

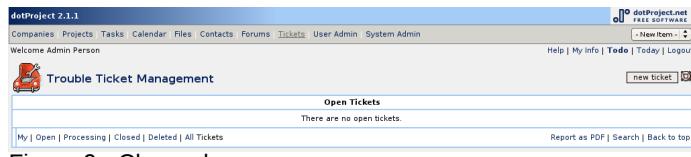


Figura 9 - Chamados

## Customização do sistema

Você poderá inserir campos, exibir ou ocultar funcionalidades, definir atributos de um usuário padrão, melhoria da tradução.

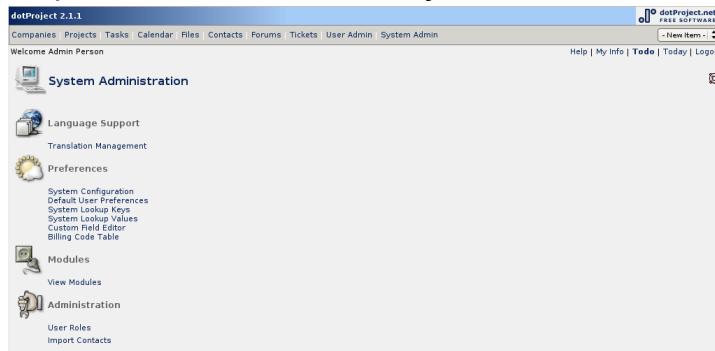


Figura 10 - Customização/administração do sistema

## Relatórios

Existem diversos relatórios disponíveis: Desempenho do usuário, Horas alocadas por usuário, estatísticas de projetos, lista de tarefas, relatório das tarefas, tarefas por usuário, Tare-

fas atrasadas por usuário, tarefas completadas na última semana, tarefas a ser completadas na próxima semana, gant, entre outros. Sendo que também é possível fazer adição de mais relatórios através de módulos Add on.



Figura 11 - Relatórios

Particularmente, gosto de soluções simples e eficazes. O dotProject tem uma navegação relativamente simples, é um sistema customizável, multiplataforma, e pode ser trabalhado em equipe. É possível fazer toda gestão de projetos, tarefas, cronogramas, recursos humanos, empresas, clientes. Portanto se você está procurando uma alternativa ao MS Project, vale a pena fazer um estudo de caso com a ferramenta opensource dotProject.



## Maiores informações:

**Site oficial dotProject:**

<http://www.dotproject.net>

**Lista em português do dotProject:**

<http://listas.softwarelivre.org/cgi-bin/mailman/listinfo/dotproject-br>



**SAYMON CASTRO** é Pós-Graduando em Gerenciamento de Projetos pela FIJ, Tecnólogo em Redes de Computadores pelo Ifes, Analista de Tecnologia da Informação pelo Ifes e professor de ensino profissionalizante (Técnico em Informática).



# Para quem ainda não conhece, apresento o OpenDocument Format - ODF

Por Jomar Silva

Na edição passada da revista, abordei o tema Padrões Abertos para garantir que todos os leitores saibam exatamente diferenciar o que é um padrão aberto e o que é uma especificação publicada na Internet. No artigo de hoje vou falar sobre um padrão aberto que está sendo adotado no mundo todo e mudando a forma pela qual armazenamos documentos editáveis de escritório.

Documentos de escritório são aqueles que utilizamos no dia a dia em nossa vida pessoal ou corporativa, tipicamente textos, planilhas e apresentações. São chamados “editáveis” pois comumente são documentos que são armazenados em forma editável, e são utilizados como base para o desenvolvimento de novos documentos, não sendo raras as

situações onde a edição do documento é feita de forma colaborativa. Existem ainda os documentos de escritório em forma final, comumente armazenados em um formato não editável, como o PDF (ok, eu sei que você pode até editar um PDF, mas o formato PDF foi criado pela indústria gráfica para armazenar documentos em sua forma final, como se fosse uma impressão eletrônica do documento).

Na década de 90, vivíamos uma situação interessante onde era crescente a utilização de uma suíte de escritórios específica (e proprietária) para a produção de documentos de escritório e tal utilização massificada (chegando a taxas superiores a 90% de liderança de mercado) acabou fazendo com que o formato de documentos utilizado por esta suíte se

tornasse o que comumente se chama de “padrão de mercado” (eu odeio este termo). Um dos problemas que isso nos trouxe é que em média a cada dois anos, o fabricante da tal suíte de escritório lançava uma nova versão da suíte, alterando o formato dos documentos gerados por ela. Isso demandava a atualização da suíte de escritório, e portanto aquisição de uma nova licença, por parte de praticamente todos os usuários, uma vez que os documentos gerados por uma nova versão da suíte não era suportado pelas versões anteriores. Bastava que uma organização em uma rede de troca de documentos fizesse o upgrade para que todas as demais acabassem tendo também que fazer a atualização, pois os problemas gerados pela falta de interoperabilidade eram elevados e o retrabalho (gravar o documen-

to na versão anterior do formato), além de improdutivo por vezes gerava a perda de parte das informações ou da formatação do documento (cá entre nós, todos sofremos com isso pelo menos uma vez na vida, não é?)

Existiam outras opções de suítes de escritório na época, mas a utilização massificada da suíte que dominou o mercado acabou forçando todas as organizações a adotá-la, pois era mais barato comprar a licença e instalar a nova suíte do que ficar o tempo todo tendo retrabalho para importar e exportar documentos.

Não foi a primeira vez na história da humanidade que a forma de armazenar informações nos trouxe problemas. Gosto de lembrar a todos em minhas palestras que a civilização egípcia utilizava uma forma muito peculiar de escrita,

os hieróglifos, e através dela documentou toda a sua história em seus monumentos e papirus. O problema é que por volta do ano 300 D.C. morreu o último egípcio que sabia ler os hieróglifos e com ele morreu a possibilidade de se utilizar as informações existentes naqueles documentos. Durante aproximadamente 1.500 anos os hieróglifos desafiavam os pesquisadores e em 1799 o exército de Napoleão encontrou no norte do Egito, na cidade de Rosetta uma pedra que ajudaria mudar esta história. Esta pedra, que ficou conhecida como a "Pedra da Rosetta", continha um mesmo texto escrito em hieróglifos, demótico e grego clássico, a única língua conhecida na época de Napoleão (para os mais curiosos, os últimos faraós do Egito, os Ptolomeus, eram de origem Grega, por isso o texto em grego gravado na pedra). Através de

engenharia reversa e partido do texto em grego, em 1822 os hieróglifos foram decifrados, iniciando assim os estudos sobre sua formação e permitindo o aprofundamento dos estudos sobre a civilização egípcia. Note que este processo só foi possível pois existia uma versão conhecida do texto, e infelizmente nos nossos tempos (ou em um futuro próximo) a engenharia reversa talvez não possa ser aplicada, pois é comum encontrarmos somente a versão eletrônica de nossos documentos (e sem uma cópia impressa, nossos herdeiros não poderão fazer muito para desvendar nossos hieróglifos digitais). No nosso caso, corremos o risco de perder uma parte da nossa própria história, pois grande parte dos documentos governamentais gerados nas últimas décadas já são, infelizmente, hieróglifos digitais.

**“ Os especialistas no acervo da Biblioteca de Alexandria consideram que se o conhecimento não tivesse se perdido pelo fogo em 646 D.C. poderíamos hoje ter a cura para doenças como o câncer.**

Jomar Silva

Ainda olhando as lições da história, o Império Romano criou uma biblioteca no norte do Egito, na cidade de Alexandria que é considerada como o maior repositório de documentos que a humanidade já criou. Durante aproximadamente 1.000 anos, todos os documentos encontrados no Império eram levados para a biblioteca, que apesar deste nome, funcionava como uma universidade aberta e seus documentos foram base para o desenvolvimento de importantes obras sobre geometria, tri-

gonometria e astronomia, idiomas, literatura e medicina. Infelizmente a biblioteca foi totalmente destruída pelo fogo em 646 D.C., queimando praticamente todo o conhecimento escrito que o império romano tinha coletado e gerado em quase 1.000 anos. Os especialistas no acervo da biblioteca consideram que se este conhecimento não tivesse se perdido, poderíamos hoje ter cura para doenças como o Câncer e alguns desenvolvimentos em tecnologia que nem sequer conseguimos imaginar.

ceu com empresas no passado, por exemplo, alguém aí lembra do WordStar ?). Para muita gente no Império Romano, imaginar a Biblioteca de Alexandria destruída pelo fogo também era um despautério, mas aconteceu.

Voltando aos nossos tempos, na década de 90, uma empresa alemã chamada Star Division desenvolvia uma suíte de escritório chamada Star Office, distribuída gratuitamente. Dentro desta empresa, foi iniciado o desenvolvimento de um formato de documentos editá-

to das informações e por isso, junto com a criação do projeto do OpenOffice.org foi criado um projeto para desenvolvimento colaborativo de um formato de documentos para ser utilizado pela suíte, com base no formato XML desenvolvido anteriormente pela Star Division.

Cabe lembrar aqui que até o final do primeiro trimestre do ano passado (2008), a especificação dos formatos proprietários da suíte de escritório dominante do mercado era fechada e de controle direto do seu fabricante. Já há alguns anos esta empresa permitia que seus parceiros acessassem esta especificação mediante a assinatura de um contrato específico e por isso, não eram abundantes as empresas que topavam esta proposta. Até então, todo o suporte aos formatos proprietários implementados em softwares livres, como o OpenOffice.org, era fruto de engenharia reversa e por este motivo, nem sempre estes softwares tratavam os documentos de forma adequada. Atualmente esta documentação está disponível na Internet, mas creio que todos aqui já sabem que ela não é um padrão aberto, é apenas uma especificação publicada.

Em 2002, foi lançado o OpenOffice.org versão 1 e o StarOffice versão 6, ambos com suporte ao formato de documentos desenvolvido (exten-

**“ É importante a garantia de que os documentos armazenados em ODF poderão ser abertos em 10, 50 ou 100 anos. ”**

Jomar Silva

Muita gente considera que utilizar um formato proprietário para armazenamento de documentos não é tão mal assim, pois existe uma grande empresa por trás deste formato. A Biblioteca de Alexandria também era assim, indestrutível, e da mesma forma que muita gente acredita que a tal empresa jamais deixará de existir (levando consigo o formato de documentos, como já aconte-

veis de escritório baseado em XML, para ser utilizado pelo software.

Em 2000 a Sun Microsystems comprou a Star Division e decidiu licenciar em software livre o código fonte da suíte, nascendo assim o projeto OpenOffice.org. Não fazia sentido algum ter uma suíte de escritório em software livre que tivesse que utilizar um formato proprietário para armazenamen-

são .sxw, alguém aí lembra disso?). Com este lançamento, estava comprovado que a utilização da tecnologia XML para armazenar documentos editáveis de escritório era viável e o novo formato criado atraiu a atenção de outras comunidades de software livre, que se envolveram no projeto de desenvolvimento do novo formato (destaque na época para o Koffice).

O novo formato atraiu a atenção da indústria e por isso foi criado em 2002 um comitê para seu desenvolvimento dentro do OASIS, uma organização internacional sem fins lucrativos que trabalha com o desenvolvimento de padrões abertos (de acordo com os critérios que vocês já conhecem, descritos na edição passada da revista).

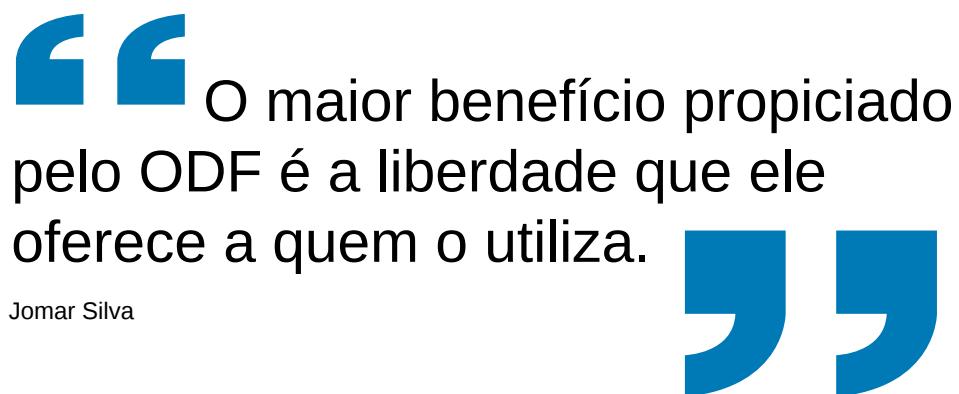
Em Dezembro de 2004 o segundo draft do padrão foi aprovado no OASIS, em Fevereiro de 2005 o terceiro draft é colocado em consulta pública e em Maio de 2005 o ODF é aprovado pelo OASIS. Note que esta aprovação ocorre após 6 anos de desenvolvimento do padrão, 5 dos quais através de um processo aberto e colaborativo.

Em Setembro do mesmo ano, o ODF é encaminhado para a ISO, e em Maio de 2006 o ODF é aprovado por unanimidade, se tornando a norma ISO/IEC 26.300, adotada (este é o termo para a aprovação de uma norma na ABNT) no Brasil

em Maio de 2008 como a NBR ISO/IEC 26.300:2008 (ou seja, o ODF é a norma técnica brasileira para documentos editáveis de escritório).

O maior benefício propiciado pelo ODF é a liberdade que ele oferece a quem o utiliza, permitindo que qualquer aplicação que suporte o padrão (atualmente mais de 50 aplicações já o fazem), possa ser utilizada

va da indústria e seu desenvolvimento não depende de uma única empresa. A segunda é que sua especificação está disponível na Internet e já existem diversas implementações de referência do padrão licenciadas em código aberto, como o OpenOffice.org. Novamente, a continuidade destas aplicações depende mais das pessoas (como eu e você) do que das empresas, mas acho que

O maior benefício propiciado pelo ODF é a liberdade que ele oferece a quem o utiliza.

Jomar Silva

para manipular os documentos, independente do sistema operacional utilizado ou da plataforma computacional (fazendo nisso, para quem gosta das buzzwords, se você é da geração "da nuvem" ou da turma do "SaaS", pode utilizar ODF no Google Docs, sem necessidade de instalar nenhuma suíte de escritório em seu computador).

É importante também a garantia de que os documentos armazenados em ODF poderão ser abertos em 10, 50 ou 100 anos, por dois motivos principais. O primeiro deles é que o ODF é uma iniciativa colaborati-

todo mundo aqui já sabe disso, não é ?

Mais do que ter a liberdade, com a utilização do ODF você permite que os outros tenham liberdade também, não obrigando ninguém a adquirir uma licença de software algum para poder trocar documentos com você. Mais legal ainda é que existindo uma suíte de escritórios tão completa como o OpenOffice.org disponível gratuitamente, você pode ensinar e recomendar a todos que o instalem em seu computador para poder editar nativamente documentos em ODF.

Muita gente deve estar se

perguntando “Por que o Jomar fala tanto em OpenOffice.org e não no BrOffice.org?”. Para quem não sabe, o OpenOffice.org e o BrOffice.org são exatamente a mesma aplicação, e basicamente o nome muda porque quando o OpenOffice.org foi lançado internacionalmente, a marca OpenOffice.org já havia sido registrada no Brasil e a empresa detentora do registro não abriu mão da sua marca. A comunidade brasileira precisou então mudar o nome da “nossa versão” traduzida do OpenOffice.org para BrOffice.org e por incrível que pareça isso nos deu alguns benefícios. O primeiro e mais importante deles é que temos hoje uma das maiores e mais ativas comunidades de OpenOffice.org do mundo e muita gente chega á esta comunidade imaginando que o software é todo desenvolvido no Brasil. Quando aprende a história que acabei de contar, o colaborador já está apaixonado pela suíte e pela comunidade e aí não consegue mais abandoná-la.

Existem outros softwares que suportam o ODF e dentre eles destaco o Lotus Symphony da IBM, o Star Office da Sun, os softwares livres Koffice e Abiword (só para citar alguns) e por incrível que pareça, no final do mês de Abril se- rá lançado o Service Pack 2 do Office 2007 da Microsoft que terá como uma das suas novidades o suporte nativo ao ODF (como escrevo este artigo an-

tes do lançamento, gostaria de deixar claro que “Só acredito vendo :”). Ainda falando em Microsoft Office, já existem plug-ins para fazer com que ele suporte ODF e destes plug-ins, o mais destacado (e até onde sei o mais completo) é desenvolvido pela Sun Microsystems e pode ser facilmente encontrado na Internet. Como se pode ver, a escolha da sua suíte de escritório é toda sua.

Atualmente participam do desenvolvimento do ODF diversas empresas de tecnologia como Novell, Google, Sun Microsystems, IBM, Red Hat e Microsoft (entre outros), além de diversos especialistas do mundo todo, como este que vos escreve (sim, tem brazuca desenvolvendo o ODF :)).

A última versão do ODF aprovada pelo OASIS é a versão 1.1, que se diferencia da versão 1.0 (a mesma da norma ISO), por contar com suporte á acessibilidade. A versão 1.2 do ODF está em fase final de desenvolvimento no OASIS e convido a todos a acompanhar nosso trabalho (basta buscar OASIS ODF TC e você nos encontra online).

Espero que todos agora saibam o que é o ODF, pois pretendo falar mais sobre ele nos próximos artigos da revista. Eu acredito que estamos com o ODF hoje no mesmo estágio que estávamos com o HTML em 1995 e por isso, ainda vai passar muita água debaixo da ponte e as aplicações

realmente inovadoras que utilizam o ODF ainda serão desenvolvidas. Meu principal objetivo hoje é fazer com que você seja o autor de uma delas :) 

## Maiores informações:

### ODF Alliance:

<http://www.odfalliance.org>

### Site do OpenOffice.org:

<http://www.openoffice.org>

### Site do BrOffice.org:

<http://www.broffice.org>

### Site da OASIS:

<http://www.oasis-open.org>

### ISO:

<http://www.iso.org>

### ABNT:

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

### Blog do Jomar:

<http://homembit.com>



**JOMAR SILVA** é engenheiro eletrônico e Diretor Geral da ODF Alliance Latin America. É também coordenador do grupo de trabalho na ABNT responsável pela adoção do ODF como norma brasileira e membro do OASIS ODF TC, o comitê internacional que desenvolve o padrão ODF (Open Document Format).



Adam Jakubia - sxc.hu

# PANÓPTICO NA REDE

Por Filipe Saraiva

“

O Panóptico de Bentham é a figura arquitetural dessa composição. O princípio é conhecido: na periferia uma construção em anel; no centro, uma torre; esta é vazada de largas janelas que se abrem sobre a face interna do anel; a construção periférica é dividida em celas, (...) elas têm duas janelas, uma para o interior, correspondendo às janelas da torre; outra, que dá para o exterior, permite que a luz atravesse a cela de lado a lado. Basta então colocar um vigia na torre central, e em cada cela trancar um louco, um doente, um condenado, um operário ou um escolar. Tantas jaulas, tantos pequenos teatros, em que cada ator está sozinho, perfeitamente individualizado e constantemente visível. O dispositivo panóptico organiza unidades espaciais que permitem ver sem parar e reconhecer imediatamente.

Michel Foucault

”

O fragmento do texto acima foi retirado do livro “Vigiar e Punir”, do francês Michel Foucault, considerado um dos maiores intelectuais do último século. Suas obras resultaram em contribuições valiosas para uma diversidade de ciências e áreas do saber, indo da psiquiatria à filosofia, pedagogia, história, política, direito... no livro citado, Foucault faz um apanhado histórico da legislação e dos métodos penais de diversos países e instituições, expondo como esta área mudou suas práticas no decorrer do tempo e foi tornando-se cada vez mais util em suas ações.

A ideia de discutir-se as implicações psicológicas e sociais, constituídas pela arquitetura de uma prisão de vigilância panóptica, não foi proposta por Foucault; porém, ele foi um dos que desenvolveu a teoria de tal forma que hoje ela é utilizada para discutir vários tipos de opressão hierarquizantes – seja nos manicômios e presídios até fábricas e escolas.

Com o tempo, a teoria sobre o Panóptico gerou um plano teórico/discursivo mais condizente com nosso atual estado de sociedade – onde a mobilidade é maior, mas a vigilância não deixa de existir. Falto sobre a Sociedade do Controle, que nos observa no nosso cotidiano, sem necessariamente nos privar da liberdade de escolha e de deslocamento.

Para mim, falar de contro-

le na Internet envolve os dois conjuntos teóricos. Em alguns momentos é mais interessante falar de Sociedade do Controle (Google Latitude?); em outros, como as recentes leis de ciber-crimes, falar de Panóptico é mais condizente com a noção hierárquica de vigilância – como um servidor que impede o acesso de internautas a determinados endereços (procure por Grokster e acesse o site para ter uma ideia disso). Para outras abordagens, acho que uma teoria híbrida seja ainda mais interessante: um servidor que guarda seus logs de acesso é tanto panóptico – porque ele está o “observando o tempo todo” - como um agente do controle – você tem liberdade de deslocamento pelo ciberspaço até certo ponto.

Os governos e grandes empresas do mundo passam agora por um momento de implementação de uma política

panóptica/controladora no ciberspaço. A vigilância está crescendo e se fortalecendo a partir de leis, tratados e produtos disponibilizados que cada vez mais nos conhecem, sabem o que fazemos e o que buscamos na rede.

Uma parte dessas leis é fruto de lobby dos conglomerados de empresas fonográficas, que tentam impedir o compartilhamento de seus bens culturais sujeitos ao copyright. Já havia comentado sobre isso em meu último texto na revista (ver Espírito Livre nº 1 – Julgamento do The Pirate Bay): existe uma pressão em parlamentos e congressos no mundo para aprovar leis que impeçam essa distribuição e, na ânsia de descobrir quem são os partilhadores, muitas dessas leis trazem artigos que violam a privacidade das pessoas – como registros de log contendo: quais sites foram vi-

**“ A vigilância está crescendo e se fortalecendo a partir de leis, tratados e produtos disponibilizados que cada vez mais nos conhecem, sabem o que fazemos e o que buscamos na rede. ”**

Filipe Saraiva

sitados, quanto tempo passou em cada um deles, sobre o que eles tratam...

Outra frente parte dos magistrados mais conservadores, que interpretam a dinâmica das redes como similar a das sociedades convencionais, tentando transpor leis e juízos de valor da realidade física (palpável) para a virtual. Daí, por exemplo, que distribuir música, filmes e livros ou até simplesmente fazer legendas para um seriado sejam considerados, respectivamente, furto e quebra de direito autoral.

E por último, outro grupo que propõe leis de controle sobre a rede são, por vezes, ligados aos movimentos de garantia dos direitos humanos que gostariam de interceptar redes de tráfico de drogas, armas e crimes bárbaros como os de pedofilia. Certamente, estas atitudes listadas causam prejuízos a sociedade como um todo; porém suspender garantias e conquistas de direito a privacidade de todos os internautas não seria aceitável – tanto como violar a correspondência de todo mundo para interceptar estas mesmas redes criminosas também não o são. Acredito que deva haver um aparato para coibir estes crimes pela rede, mas as leis que versariam sobre esta questão não deveriam colocar a totalidade dos internautas como suspeitos destes atos.

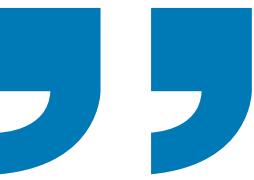
Para alguns exemplos destas leis invasivas, temos visto

recentemente várias delas transmitindo em diferentes países – vale citar alguns, por exemplo, França, Coreia do Sul, Grã-Bretanha e Brasil.

Na França, existe um projeto de resposta gradual que suspenderá o acesso a Internet de partilhadores por até um ano (ver Espírito Livre nº 1 – Julgamento do The Pirate Bay). A responsabilidade por acusar e investigar os internau-

ce administradores de redes sociais como Facebook e MySpace a fornecer detalhes de amigos e contatos de usuários. O governo diz que o projeto é necessário para combater o terrorismo no país, e que para isso é fundamental ter acesso irrestrito a esses dados.

No Brasil temos a Lei Azevedo que, através de seus artigos bastante abertos a várias interpretações, criminaliza o

  
Acredito que deva haver um aparato para coibir estes crimes pela rede, mas as leis que versariam sobre esta questão não deveriam colocar a totalidade dos internautas como suspeitos destes atos.  


Filipe Saraiva

tas compartilhadores de bens culturais ficaria a cargo de uma entidade chamada HADO-PI. O projeto na Coreia do Sul é parecido com o francês, exceto que o poder de encerrar qualquer atividade online, seja site ou acesso a rede, seria concorrente ao Ministério da Cultura, Desporto e Turismo.

A Grã-Bretanha estuda a possibilidade de uma lei que for-

compartilhamento p2p, a programação de eletrônicos buscando-se novos usos, coloca servidores para gravarem logs de acesso à rede... dentre outros. Importante que temos um movimento muito forte para barrar esta lei, que já contou com debates, manifestações e uma extensa petição online ainda no ar.

Quanto aos produtos dis-



Cabe a nós lançar um olhar crítico sobre o tema, e não apenas tratá-lo como mania de perseguição ou neurose...



Filipe Saraiva

ponibilizados que favorecem a implementação do panóptico/controle na Internet, podemos citar os bastante utilizados – até por este que vos escreve – serviços do Google. Buscador, Gmail, Blogger, Youtube, Orkut... todos eles se apropriam de informações pessoais nossas – cedidas espontaneamente, vale lembrar – que são utilizadas pelos bots da empresa para dispor de propaganda condizente com nossas preferências. Isso é também uma forma de vigilância, uma maneira de termos informações pessoais disponibilizadas para alguém “estranho”, que conhece nossa intimidade através de nossas correspondências eletrônicas, posts em blogs pessoais, buscas...

Um interessante vídeo ficção-documentário sobre o Google e a privacidade de seus clientes pode ser encontrada em EPIC 2014. Algumas ponderações feitas nele podem assustar a aqueles que nunca pensaram no impacto de disponibilizar informações pessoais para a empresa.

Estamos diante de uma era em que a vigilância está se fortalecendo na rede. Não apenas aos partilhadores de bens culturais, mas até pessoas que utilizam webmail, blogs grátis e redes sociais estão cada vez mais tendo suas informações repassadas a desconhecidos ou sendo vítimas do olhar de um “Big Brother” que quer saber – ou já sabe – qual o conteúdo daquele pacote roteado para a sua máquina. E cabe a

nós lançar um olhar crítico sobre o tema, e não apenas tratá-lo como mania de perseguição ou neurose daqueles que o discutem, como se dá em reações normalmente encontradas em alguns sites ou listas de discussão por aí. 

### **Maiores informações:**

#### **Grokster:**

<http://www.grokster.com>

#### **Documentário Ficção EPIC 2014**

<http://www.youtube.com/watch?v=4OZ-ANCEchM>

#### **Petição contra a Lei Azeredo**

<http://www.petitiononline.com/veto2008/petition.html>

#### **Site Liberdade na Fronteira**

<http://www.liberdadenafrenteira.blogspot.com>



**FILIPE DE OLIVEIRA SARAIVA** estuda Ciências da Computação na Universidade Federal do Piauí, entusiasta do GNU/Linux, da Cultura Livre e das possibilidades de criação coletiva oferecidas pelo mundo conectado. É pesquisador da área de Cibercultura, Pesquisa Operacional e Inteligência Artificial.

## **IV EVIDOSOL e I CILTEC Online**

**Encontro Virtual de Documentação em Software Livre e Congresso Internacional de Linguagem e Tecnologia Online - 3 a 5 de junho de 2009**



# Fazendo as pazes com o bicho-papão: A matemática e o SL

Por Sinara Duarte

A matemática, para a maioria, das pessoas, é o bicho-papão, pois é a disciplina que mais apavora, que mais reprova. Estima-se que em um universo de cem reparações, 70% por cento, aproximadamente, são atribuídas a essa disciplina, o que demonstra o fracasso do ensino da matemática nas instituições escolares brasileiras.

O reflexo da precariedade do ensino brasileiro foi mostrado no PISA – Programa Internacional de Avaliação de Alunos (2000) – onde os estudantes brasileiros ficaram na última colocação na prova de matemática concorrendo com outros trinta países.

Na escola a matemática é vista como vilã, ao mesmo tempo que eterniza os gênios. Todo mundo lembra da infância, daquele aluno, com cara

de nerd, que sabia tudo de matemática, fazendo “ares de doutor” sem ter sequer completado o ensino fundamental (antigo 1º grau). Infelizmente ainda hoje, predomina no ideário coletivo, que a aprendizagem da matemática é para poucos afortunados que nasceram com este dom, quando na verdade, já nascemos programados para aprender matemática. Como assim?

Que a matemática, está em todo lugar, isso já sabemos, não apenas na fatura do cartão de crédito, no desconto do Imposto de Renda, no alinhamento das rodas de seu carro, mas indiretamente, também fazemos cálculos inconscientemente. Mesmo até os mais avessos a matemática, utiliza essa habilidade corriqueiramente sem perceber.

Exemplificando: Quem te-

ve a dádiva de ser pai ou mãe, ou mesmo ficou observando uma criança dar seus primeiros passos, percebe a hesitação da mesma, quando encontra um obstáculo em seu caminho: uma poça de água ou uma calçada mais alta. Neste momento, a criança hesita, pois seu cérebro ainda em formação, enfrenta uma situação nova na qual não está habituada. No caso de cérebro adulto, este é capaz dentro de uma fração de segundos, como uma poderosa máquina de calcular, relacionar a distância a partir de seu campo de visão (comparação) até o obstáculo, calcular o ângulo (trigonometria) necessário para dar o passo, calcular a força necessária para a impulsão (física), o risco envolvido (probabilidade: eu caio ou não?) para por fim, definir qual a melhor estratégia: dar a volta ou tentar pular a po-

ça de água assim mesmo.

Neste simples exemplo comum, a maioria dos humanos, que é o ato de caminhar, nós temos diversas habilidades matemáticas que são requeridas, que no infante, ainda não são tão bem desenvolvidas, daí as quedas e tombos constantes na primeira infância. Com o tempo, após alguns erros, acertos e cicatrizes no joelho, a criança já comprehende e guarda na memória essas "habilidades matemáticas", deixando pais e avós menos aflitos.

Outro exemplo, são os feirantes que pouco frequentaram as escolas tradicionais, mas sabem "de cor" fazer cálculos envolvendo álgebra, percentagem e aritmética. É na luta pela sobrevivência que tais pessoas desenvolvem formas específicas de raciocínio matemático.

Se a matemática é algo tão simples, até mesmo as populações sub-escolarizadas, porque nas escolas, ouvimos tanto o discurso: "Eu odeio a matemática" ou "Eu não consigo aprender matemática".

A maioria dos alunos, trazem consigo concepções errôneas a respeito do ensino da matemática, recheadas de medos e tabus. Não é objetivo deste artigo, debater as causas desse fracasso até porque já existe na literatura, bastante material a esse respeito, de artigos a teses de doutorado. Nossa intuito é apresentar a pais e educadores uma alternativa, visto que o mundo livre apresenta excelentes softwares de apoio à pesquisa e ao ensino da Matemática. Todavia, no Brasil, a utilização da tecnologia livre, principalmente nas escolas, ainda é modesta, na maioria das

vezes, por falta de divulgação.

Assim, o objetivo deste artigo além de advogar em prol da matemática, redimindo sua culpa, é também difundir o uso de ferramentas livres no ensino da matemática para crianças no ensino fundamental. Noutra oportunidade falaremos de softwares livres que auxiliam para a aprendizagem da matemática para alunos do ensino médio e acadêmicos.

Na infância, uma das principais dificuldades é a aprendizagem da tabuada. A maioria dos alunos até que comprehende as operações fundamentais (adição, subtração, multiplicação e divisão) com unidades e dezenas sem muitos problemas, mas tem dificuldade em memorizar a tabuada.

Há poucos anos atrás, era comum os professores utilizarem palmatórias para dar "bolos" nas mãos daqueles que não sabiam a tabuada na ponta da língua. Atualmente, o instrumento de tortura mudou, mas a prática continua se perpetuando em sala de aula, por meio de "chapéus de burro" ou sabatinas.

A sabatina é a prática, diga-se inútil, de cobrar a aprendizagem da tabuada oralmente, de traz para frente, de frente para traz, expondo muitas vezes, a criança ao ridículo quando esta erra. Muitas vezes, a criança até sabe, mas o medo de estar diante de um público (seus amigos), a pres-

**“Há poucos anos atrás, era comum os professores utilizarem palmatórias para dar “bolos” nas mãos daqueles que não sabiam a tabuada na ponta da língua.”**

Sinara Duarte



Figura 1: Tela principal do Tuxmath

são do tempo e a cobrança do professor fazem com que o aluno se desespere (o famoso brancão) e faça da matemática, um bicho de sete cabeças, ops, quer dizer de 10 cabeças.

Não existe uma opinião unânime no meio educacional. Uns são contra a memorização da tabuada, outros são a favor. Particularmente, acredito que é fundamental compreender a tabuada, visto que se souberem a “maledita” de cor, poderão ser mais ágeis ao resolver as situações-problema envolvendo principalmente a multiplicação e divisão. Uma vez compreendido, a tabuada pode, aos poucos, ser memorizada sem traumas. O que não se pode admitir, em pleno século XXI, é o aluno ficar recitando uma ladainha de números, fórmulas, sem terem entendido o significado.

Com o advento da Infor-

tem e educam ao mesmo tempo. Como o espaço é curto, vamos nos ater a apenas um software por mês, mas esse tema dado continuidade posteriormente.

Um software capaz de divertir e ao mesmo tempo ensinar é o Tux math of Command, ou simplesmente, Tuxmath pertencente ao projeto Tux4kids (<http://tux4kids.alioth.debian.org/index.php>). Este software educativo apresenta o mesmo estilo do clássico jogo arcade para Atari, Missile Command, (se tiver mais de trinta anos vai lembrar) que tornou-se febre nos anos 80. O objetivo do jogo é impedir que as bom-

mática Educativa aliada ao movimento do Software Livre, desenvolvedores de todo mundo começaram a programar específicos para o ensino da matemática, direcionada para o público infantil que recriam, diver-

bas atingiam as construções de uma cidade sob ataque.

No Tuxmath o raciocínio é o mesmo, só que as bombas são na verdade “operações matemáticas” que serão destruídas somente se o jogador acertar o cálculo. Ao aparecer a bomba, ou melhor, o problema a criança necessita digitar o número correspondente a resposta para se defender. Por exemplo  $4 \times 3$ , digita-se a resposta (12) e tecla o ENTER. Como a resposta está correta, dispara-se um feixe de laser destruindo-a. Caso estivesse errada, o raio vai em outra direção, e sinto muito, será uma construção a menos no planeta Linux, deixando o Tux bem chateado (figura 2 e 3).

Da mesma forma do videogame, a dificuldade vai aumentando à medida que o jogador passa para um nível superior. A cada vitória mudam-se os cenários, a velocidade e a quantidade das continhas.



Figura 2: Tuxmath em ação



Figura 3: Tuxmath em ação... e o tux furioso...

Não existe a possibilidade de pausar o jogo, nem controle de velocidade apenas no módulo de treinamento. Aciona-se a pausa na letra P.

No treinamento é possível escolher entre tabuada de somar, subtrair, multiplicar ou dividir, inclusive desde a versão 1.5.8 já vêm a possibilidade de trabalhar com números inteiros (números negativos e positivos), conteúdo das séries terminais. Ressalta-se que já saiu a versão 1.7.0 (beta).

Os níveis de dificuldade são divididos por patentes. Do cadete especial, nível mais fácil ao especialista, nível mais aprofundado. Para finalizar usa-se o ESC. Apesar de bem simples, atrai as crianças e adultos por seu visual e sons que lembra um game. Existe também um hall da fama, com um ranking dos melhores jogadores.

A versão estudada (1.5.8) não apresenta muitas funcionali-

dades, nem apresentou bugs. Agora um aspecto que merece ser destacado é a ausência de violência. Mesmo perdendo o jogo, ou seja, tendo seu iglu bombardeado, os amigos do Tux não morrem, apenas se retiram calmamente sem demonstrações sanguinolentas de chacina animal. (O Greenpeace agradece!) E a medida que o aluno vai acertando, existe a possibilidade dos amigos do Tux serem repatriados e ganhar novamente seu iglu.

Sob licença GPL, este software já foi traduzido para o português do Brasil (PT\_BR) além de outras línguas como espanhol, francês, árabe, dentre outras. Possui versão para Linux, Windows e MacOSX. Ide-

al para crianças de 6 a 12 anos, que já compreendem os conceitos aritméticos, mas apresentam dificuldade na memorização da tabuada.

Esse ano, Tuxmath quanto seus fraternal Tuxpaint e Tuxtype foram classificados para participar do Google Summer of Code, um programa que paga bolsas (US\$ 4000) para estudantes do mundo todo, trabalharem em projetos de código aberto e software livre durante as férias do verão americano (daí o nome). Sugestões e melhorias para estes softwares podem ser enviadas pelo site oficial do evento. (<http://googlesummerofcode.blogspot.com>)

## Instalando o Tuxmath

O Tuxmath pode ser instalado facilmente via synaptic e/ou nas distribuições Debian/Ubuntu pelo comando: apt-get install tuxmath. Outra vantagem do tuxmath, é sua leveza, consome apenas 5931Kb na hora da instalação.

“ É importante destacar que o uso de jogos computacionais na educação formal, não substitui as aulas convencionais... ”

Sinara Duarte

Enfim, como vimos o Tuxmath, é apenas um dentre dezenas de softwares que podem ser utilizados na educação matemática.

Quem gosta do tema e não quiser esperar pelos próximos números, pode conhecer o trabalho de conclusão do curso do Professor Ricardo Pinheiro. Sua monografia com o título "Software livre e Matemática: Opções de pesquisa e ensino apresentou uma pesquisa de mais de 100 softwares livres para o ensino da matemática. Ta muito organizado, bem escrito, numa linguagem clara, vale a pena ler. Seu trabalho na íntegra está hospedado no seu blog pessoal Estudio da Introspecção (<http://estudiodaintrospeccao.blogspot.com>) no qual também disponibiliza a monografia completa no formato PDF.

Enfim, o mundo do software livre apresenta diversas possibilidades educativas de utilização do computador na educação. Aqui apresentou-se apenas a ponta do iceberg, pois a cada dia, cresce o interesse do público em geral por jogos computacionais, e por que não aliar o útil ao agradável? Por que não aliar conhecimento com divertimento? Por que a matemática tem que ser chata? Por que não utilizar o software livre na educação?

É notório que o uso do computador aumenta o interesse dos alunos tornando-se mais concentrados, mais receptivos, curiosos, tendo prazer em realizar as atividades escolares. Todavia, é importante destacar que o uso de jogos computacionais na educação formal, não substitui as aulas convencionais, mas complementa o ensino dos conceitos teóricos de Matemática aprendidos nas salas de aulas.

Ensinar Matemática é muito mais do que apenas decorar fórmulas e resolver problemas, é desenvolver nos educando, o raciocínio lógico, estimulando o pensamento abstrato, a concentração e a curiosidade. Compete a nós, educadores, pais, estudiosos, curiosos desse assunto procurar alternativas para aumentar a motivação para a aprendizagem mais principalmente incentivar em nossas crianças o espírito e o gosto pela liberdade!

## Maiores informações:

**Projeto Software Livre**

**Eduacional:**

<http://sleducacional.org>

**Blog Software Livre na Educação:**

<http://softwarelivrenaeducacao.wordpress.com>

**Blog Estúdio da Introspecção:**

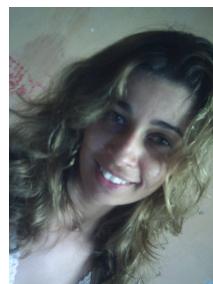
<http://estudiodaintrospeccao.blogspot.com>

**Site Oficial Google Summer of Code:**

<http://googlesummerofcode.blogspot.com>

**Site Projeto Tux4Kids:**

<http://tux4kids.alioth.debian.org/index.php>



**SINARA DUARTE** é professora da rede municipal de Fortaleza, pedagoga, especialista em Informática Educativa e Mídias em Educação, com ênfase no Software livre. Colaboradora do Projeto Software Livre Educacional e mantenedora do Blog Software Livre na Educação.

**LATINOWARE 2009**  
de 22 a 24 de outubro no Parque Tecnológico Itaipu - PTI  
Foz do Iguaçu PR | Brasil



# Festival Latino-americano de Instalação de Software Livre

## VITÓRIA/ES

Por João Fernando Costa Júnior

No sábado, dia 25/04/2009, aconteceu o FLISOL - Festival Latino Americano de Instalação de Software Livre, um evento de proporções continentais, que aconteceu em diversas cidades em toda a América Latina. Um grande evento de divulgação, instalação, informação e contribuição com o Software Livre.

No Espírito Santo, a capital Vitória foi sede de um ponto do FLISOL, organizado pela Iniciativa Espírito Livre, e que contou com a colaboração de Almir Mendes, Membro do PHP-ES e Linux-ES, entre outros. O FLISOL 2009 movimentou vários tipos de usuários, incluindo os próprios alunos da FAESA, que foram

direcionados para o evento pelos seus próprios professores.

O evento foi aberto por Almir Mendes que apresentou os palestrantes e logo a seguir chamou Guilherme Chaves que iniciou o ciclo de palestras falando sobre o uso do Linux em ambiente desktop e direcionado para usuário final. Guilherme ainda que é responsável pela matéria sobre Bluepad desta edição. Almir Mendes retorna, desta vez falando os diversos sabores de GNU/Linux, desde para os experientes aos que estão iniciando neste novo mundo. Jean (Índio) e Diego Nogueira continuam o evento falando sobre UML - User Mode Linux, um assunto que



Figura 1: Participantes marcam presença durante palestras

chamou atenção de todos os presentes por se tratar de algo não muito comum de se ver nos eventos. Lázaro Reinã, que está falando de LUA nesta edição, fechou o evento falando sobre a linguagem LUA, um assunto igualmente interessante.

O installfest não foi tão movimentado como nos anos anteriores, entretanto um fato que chamou atenção foi a instalação em um eeePC, que foi levado por um dos participantes. Segundo ele, o equipamento estava com um sistema operacional proprietário, entretanto estava muito lento e ele não estava satisfeito. Após a instalação de uma versão remix do Ubuntu no equipamento, o dono do equipamento ficou super feliz e

rindo a toa com a velocidade percebida logo de cara e com os efeitos que vinham habilitados por padrão.

Houve ainda um outro caso de um laptop que também continha um sistema operacional proprietário e que foi

imediatamente retirado do equipamento. Este último ainda há a ressalva que o mesmo havia vindo de fábrica com o sistema já instalado.

O  
FLISOL 2009  
foi realizado

com o apoio da FAESA, que nesta edição nos cedeu o auditório, bem como toda a infra-estrutura necessária para a realização do evento. Nos sinceros agradecimentos, pois sem essa ajuda possivelmente a realização do mesmo não seria possível.

Fotos do evento, bem como os arquivos utilizados pelos palestrantes podem ser encontrados no endereço <http://flisol.espiritolivre.org>.

### Entrega dos alimentos

Assim como foi feito no DFD'09, os alimentos arrecadados no FLISOL 2009 foram doados, porém desta vez no mesmo dia do evento. A instituição que os recebeu chama-se Associação Amor e Vida, que tem como uma de suas unidades no bairro Jardim América, no município de Cariacica/ES.



Figura 2: Ao final do evento, pose pra foto...

A arrecadação também rendeu mais frutos que no último evento. Desta vez entregamos aproximadamente 60 kilos de alimentos, muitos deles necessitados pela associação como relatou a responsável, Creuza, que nos recebeu prontamente, assim que ligamos para ela. Um relato que nos comoveu foi o fato dela ter dito que naquele mesmo dia havia dividido o feijão (8 KG) que a unidade tinha com a segunda unidade, que fica localizada em um outro bairro do mesmo município, ficando apenas 4 KG para cada uma das unidades. Segundo ela, aqueles alimentos do eventos haviam caído do céu. Neste momento, toda a equipe que estava responsável por entregar os alimentos sentiu que o FLISOL 2009 havia ajudado não somente os participantes que permaneceram durante parte do sábado assimilando novos conceitos e ensinamentos. Mas também havíamos feito a diferença com esta colaboração à Amor e Vida.

Agradecemos aos participantes do FLISOL 2009 em Vitória/ES, pois esta doação apesar de ter sido intermediada pela Iniciativa Espírito Livre, ela foi feita inteiramente por vocês, que fizeram suas inscrições doando os alimentos. Nossa muito obrigado a vocês e nos vemos em breve.



Figura 3: Entrega dos alimentos

### Novos projetos junto a Associação Amor e Vida

A Iniciativa Espírito Livre, juntamente com integrantes de outros grupos envolvidos com software livre no Espírito Santo, perceberam a necessidade da Associação Amor e Vida não dispor de um site da instituição, que neste contexto se torna um canal de contato entre outras pessoas que queiram ajudar, bem como uma forma da associação mostrar para a sociedade como é feito seu trabalho. Assim, a proposta de desenvolvimento de um site ficou de ser vista, bem como a manutenção e hospedagem do serviços em nossos servidores. Percebemos que isto a princípio não é nada, entretanto pode ajudar muita gente. 

### Maiores informações:

**Iniciativa Espírito Livre:**  
<http://www.espiritolivre.org/>

**FLISOL 2009 no Espírito Santo:**  
<http://flisol.espiritolivre.org>

**FLISOL - Site Oficial**  
<http://flisol.info>

**DFD'09 no Espírito Santo:**  
<http://dfd.espiritolivre.org>



**JOÃO FERNANDO COSTA JÚNIOR** é o líder do GUBrO-ES e responsável pela Iniciativa Espírito Livre / Revista Espírito Livre.



2009



## Festival Latino-americano de Instalação de Software Livre

### GOIÂNIA/GO

Por Andressa Martins

Em Goiás, a capital Goiânia também foi sede do FLISOL[0], organizado pelo Projeto de Software Livre de Goiás[1] e o Apoio da ASL-GO[2] (Associação de Software Livre).

O FLISOL 2009 movimentou vários de usuários, incluindo jovens, alunos de ensino médio e universitários direcionados para o evento pelos seus próprios professores.

Contou com o ciclo de 43 palestras, Batismo Digital coordenado pelos professores Marcelo Akira Inuzuka e Wendell Bento Geraldes e seus alunos, e hacklabs..

Chamou atenção de todos os presentes pela quantidade de pessoas, houveram 2411 pré-inscritos e 1449 pre-

sentes, o maior evento de Software Livre em número de pessoas no Estado.

Coordenado por Andressa Martins, membro dos projetos Robótica Livre de Goiás[3] e ASL-GO, contou com a ajuda dos grupos de usuários de tecnologias livres do Estado, sendo eles:

GOJava[4],  
GOPHP[5],  
GOPython[6],  
GUD-GO  
(Grupo de Usuários Debian de Goiás)[7],

GUS-GO (Grupo de Usuários Slackware de Goiás)[8], Ruby-GO[9], Rails-GO[10], Robótica Livre de Goiás, SLOG (Software Livre do Oeste Goiano)[11] e TIM-GO (Tecnologia de Informação para Mulheres de Goiás)[12]



Figura 1: Batismo Digital



Figura 2: Corredores abarrotados

O installfest não foi tão movimentado, contou com a ajuda do professor da instituição que cedeu o local para o evento, e foram realizadas 37 instalações, e feita a doação de 400 distribuições.

Foi realizado com o apoio do SENAI que cedeu o auditório, anfiteatro, 2 salas e dois laboratórios com 50 lugares. E devido ao grande número de presentes, foi necessário que a instituição cedesse mais duas salas e palestrantes foram encontrados na última hora para palestrar.

São iniciativas como o FLISOL que faz a comunidade Software Livre crescer, "E a força dela está na colaboração entre as redes-comunidades, formadas por faculdades, os grupos de usuário, os vizinhos, as empresas, etc. Grande parte da mobilização foi realizada por elas. Muito boca-a-boca, mensagens e tuitadas ocorreram pelas redes. O evento se auto-divulga pois todos

que divulgam para seus colegas. Foi feita uma divulgação em cartazes que rapidamente se espalharam pela cidade, e em algumas faculdade havia até a programação do evento impressa nos murais.

Enfim, o FLISOL foi um evento democrático, realizado pela comunidade para a comunidade, onde a oportunidade de participação é cada vez mais aberta para todos. Uma grande festa de compartilhamento de conhecimento, inclusão digital e democratização da informática.

O FLISOL é o maior evento de divulgação de Software Livre da América Latina. Ele acontece desde 2005 e seu principal objetivo é promover o uso de software livre, apresentando sua filosofia, seu alcance, avanços e desenvolvimento ao público em geral, através de installfests, palestras, entre outros.

se sentem responsáveis por ele", diz o professor Marcelo Akira.

É bom citar que tem vários membros dos grupos que autonomamente divulgam o evento em suas faculdades. Muitos são anônimos

### Maiores informações:

[0] **FLISOL 2009 em Goiás**  
<http://flisol.aslgo.org.br/2009>

[1] **Site PSL-GO**  
<http://wiki.softwarelivre.org/PSLGO>

[2] **Site ASLGO**  
<http://aslgo.org.br/>

[3] **Site Robótica Livre**  
<http://roboticalivre.aslgo.org.br/>

[4] **Site GoJava**  
<http://www.gojava.org/>

[5] **Site GoPHP**  
<http://www.gophp.com.br/>

[6] **Grupo PythonGO**  
<http://groups.google.com.br/group/pythonngo>

[7] **Site Debian-GO**  
<http://www.debian-go.org/>

[8] **Grupo GUS-GO**  
<http://groups.google.com.br/group/gus-go>

[9] **Grupo RubyGO**  
<http://groups.google.com.br/group/rubygo>

[10] **Grupo Rails-GO**  
<http://groups.google.com/group/rails-go>

[11] **Site SLOG**  
<http://slog.org.br/>

[12] **Grupo TIM-GO**  
<http://groups.google.com.br/group/tim-go>



**ANDRESSA MARTINS** é entusiasta de software livre, idealizadora dos projetos Robótica Livre, Membro da Associação de Software Livre de Goiás e curiosa.

# DIA LIVRE: IDÉIAS SIMPLES E AÇÕES PRÁTICAS

Por David A. Ferreira

Asif Akbar - sxc.hu

Tudo começou por volta do ano de 2006, com uma idéia proposta inicialmente por membro novato, em uma das inúmeras reuniões do PSL-CE (Projeto de Software Livre Ceará) para organização do III FCSL (Fórum Cearense de Software Livre).

A idéia surgiu como uma solução para os recorrentes problemas de divulgação das iniciativas do PSL-CE, a qual basicamente consistia da realização de pequenos eventos mensais, com o intuito de conseguir agregar novos colaboradores para o Software Livre e bem como seu foco principal que seria a divulgação do FCSL. Porém, infelizmente a idéia não ganhou a força necessária para ser posta em práti-

ca, principalmente devido às famosas batalhas campais que precedem os principais eventos de Software Livre no país, onde geralmente um pequeno grupo com interesses conflitantes armam palco para guerras intermináveis de questões que passam de puritanas e filosóficas, para questões capitalistas x socialistas, técnicas e existencialistas, além das famosas autopromoções em campanhas eleitorais. Tais ações acabam gerando um desperdício de forças tão grande que atrapalham o bom funcionamento da comunidade, o que contribui para formação de uma má impressão, as quais são vistas como inúteis, pois são povoadas de seres (popular: troll) briguentos que só querem fazer algazarra, e acabar com a cre-



Figura 1: Participantes concentrados no evento

dibilidade das pessoas que lutam pela legitimidade do Software Livre.

Os anos foram passando, até que em meados de 2009, em uma das reuniões do GUD-BR-CE (antiga Debian-CE), num clima super informal (shopping), contando apenas com a presença de pessoas com ações legítimas a favor do Software Livre, a idéia foi resgatada por Alexandre Alencar (Skarmeth) e apoiada por todos. Assim João Neto (Tio João) e Alexandre, resolveram unir forças, para dar corpo e voz aquela tímida idéia. Nasceu, então o Dia Livre, um evento da comunidade para a comunidade.

O Dia Livre, é um evento mensal, no qual membros da comunidade dedicam algumas horas de seu dia, para falar so-

bre Software Livre, compartilhando o conhecimento adquirido de forma gratuita e livre para todos aqueles que tem interesse em ouvir, e assim podemos ver como de fato o Dia Livre é um evento simples na forma de sua execução, porém uma ferramenta poderosa de divulgação e agregação de novos colaboradores.

No dia 18 de abril em Fortaleza, foi realizado na Faculdade Integrada do Ceará (FIC) o 2º Dia Livre, o qual contou com a participação de 30 pessoas prestigiando as palestras: Introdução ao GNU/Linux (João Neto); Desenvolvimento de Jogos com Software Livre (David Ferreira); Gestor de TI e Software Livre (Alexandre Alencar). O evento prendeu à atenção de todos, diversos causos e fatos fo-

ram relatados pelos palestrantes, durante suas jornadas com Software Livre e assim, as pessoas presentes ficaram até o final de todas as palestras, onde havia muita interação entre o público e os palestrantes, muitas perguntas foram feitas, algumas vezes surgiam comentários, os quais eram altamente pertinentes aos tópicos, agregando valor e conhecimento a todos os presentes.

Ao final do evento, as pessoas se reuniram por ainda quase trinta minutos em um bate papo descontraído, onde pediam mais informações sobre cursos, livros e certificações, além é claro das eventuais cópias de softwares como do CDLivre e das distribuições Debian e Ubuntu.

Por justamente ser algo simples de ser realizado, gostaríamos de ver iniciativas como esta espalhadas pelos quatro cantos do país, para que possamos assim mostrar para as pessoas, que podemos criar uma nação incluída digitalmente, que respeite os padrões abertos e que seja capaz de criar tecnologias nacionais respeitando a liberdade de seus usuários. Tudo isso é possível apenas com uma simples ação, o compartilhamento de conhecimento e a boa vontade.

Caso você tenha interesse em criar o Dia Livre em sua cidade, sinta-se livre para seguir ou adaptar algumas das orientações:

### Palestras

Busque em sua cidade por pessoas que tenham afinidade com Software Livre ou GNU/Linux e que de fato tenham ações condizentes com as palavras, ou seja, tome cuidado para não achar um Administrador de Sistemas, que nunca trabalhou com um sistema GNU/Linux, para justamente palestrar sobre o mesmo, afinal que credibilidade e respeito o público daria para uma pessoa que não tem vivência para poder falar de pontos fortes ou fracos deste ambiente.

Selecione por volta de duas a três palestras, variando os temas, por exemplo: coloque uma palestra básica sobre introdução ao GNU/Linux, e outras mais avançadas como desenvolvimento ou administração de sistemas, porém sempre lembrando de equilibrar o perfil do público e o tema, tente evitar a limitação de perfis, ou seja, permita que todos possam participar do evento, afinal o Software Livre é feito pela união e não pela segregação.

### Local

Procure por instituições que tenham ou desejem ter um relacionamento próximo com a comunidade de Software Livre, geralmente são faculdades ou escolas com centros de tecnologia que prezam e respeitam o crescimento de seus alunos.



Figura 2: Da esquerda para direita: João Neto, Alexandre Alencar e David Ferreira

Desta instituição necessita-se que a mesma disponibilize uma sala ou auditório para aproximadamente 40 pessoas, um projetor multimídia com computador, o qual tenha instalado o BrOffice.org/OpenOffice.org para exibir as palestras, é desejável que este computador possua alguma distribuição GNU/Linux, para caso do palestrante deseje mostrar algum software.

Preocupe-se também com a localização da instituição, já que um local com estacionamento, e de fácil acesso as linhas de ônibus é mais propício a participação das pessoas. Lembre-se para melhor expandir o alcance dos participantes procure sempre novas instituições em lugares diferentes na cidade, isso irá permitir que novas pessoas possam co-

nhecer o evento e consequentemente o Software Livre. 

### Maiores informações:

#### Grupo de Usuários Debian-CE:

<http://www.debian-ce.org/>



DAVID ALMEIDA FERREIRA  
<http://www.davidferreira.com.br>

Especialista em Engenharia de Software com ênfase em Padrões de Software(UECE), É Arquiteto de Software do Banco do Nordeste do Brasil (BNB). É integrante do Projeto de Software Livre Ceará(PSL-CE) e do Grupo de Desenvolvedores de Jogos do Ceará(GDJCE).

# AGENDA

## MAIO

Evento: **XIX SEMAC 2009**

Data: 11 e 15/05/2009

Local: São Paulo/SP

Evento: **Brazil Scrum Gathering**

Data: 12/05/2009

Local: São Paulo/SP

Evento: **Java@TV Digital**

Data: 14/05/2009

Local: São Paulo/SP

Evento: **IX Encontro de Informática da FARN**

Data: 19 e 20/05/2009

Local: Natal/RN

Evento: **Brazilian Road Show 2009 - Java EE Open Source**

Data: 20/05/2009

Local: Porto Alegre/RS

Evento: **Engenharia Software Conference**

Data: 22 e 23/05/2009

Local: São Paulo/SP

Evento: **Seminário O Novo Mercado de TI - Oportunidades Especializações**

Data: 23/05/2009

Local: Porto Alegre/RS

Evento: **Hora Livre - Por que programar em PHP?**

Data: 23/05/2009

Local: Campo Grande/MT

Evento: **Falando em Java 2009**

Data: 24/05/2009

Local: São Paulo/SP

Evento: **Brazilian Road Show 2009 - Java EE Open Source**

Data: 27/05/2009

Local: Rio de Janeiro/RJ

Evento: **Palestra sobre Desenvolvimento Aplicativos Web**

Data: 28/05/2009

Local: Vitória/ES

Evento: **Free Software Bahia 2009**

Data: 28 e 29/05/2009

Local: Salvador/BA

Evento: **III ENSL e IV Festival**

Data: 29 e 30/05/2009

Local: Salvador/BA

Evento: **1º Workshop PHP-ES**

Data: 30/05/2009

Local: Vila Velha/ES

## JUNHO

Evento: **IV EvidoSOL e I CILTEC Online**

Data: 03 a 05/06/2009

Local: Encontro Virtual

Evento: **I Encontro de CakePHP**

Data: 13/06/2009

Local: São Paulo/SP

Evento: **III ENSOL - Encontro de Software Livre da Paraíba**

Data: 19 a 21/06/2009

Local: João Pessoa/PB

Evento: **CMS Brasil 2009**

Data: 20/06/2009

Local: São Paulo/SP

Evento: **WordCamp Brasil**

Data: 21/06/2009

Local: São Paulo/SP

Evento: **10º Fórum Internacional Software Livre - FISL 10**

Data: 24 a 27/06/2009

Local: Porto Alegre/RS

Evento: **Google Developer Days 2009**

Data: 29/06/2009

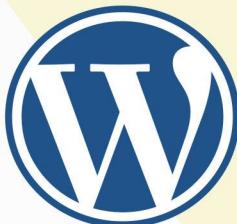
Local: São Paulo/SP



[www.cmsbrasil2009.com](http://www.cmsbrasil2009.com)

### 3 CMS's, 550 participantes

3 palestras Internacionais;  
24 atividades extras;  
Oficinas com alto nível técnico;  
Participação ativa das comunidades nacionais



#### Matt Mullenweg

Criador do **Wordpress**



#### Anthony Ferrara

Coordenador mundial  
de desenvolvimento  
do projeto **Joomla!**



#### Addison Berry

Líder do time de documentação,  
há 3 anos no projeto **Drupal**.

**20/06/2009**

Hotel Novotel Jaraguá  
São Paulo - Brasil

Realização



### INSCREVA-SE

acesse o site para mais informações  
[www.cmsbrasil2009.com](http://www.cmsbrasil2009.com)

Patrocínio





# fisl10

## 10º Fórum Internacional Software Livre

A tecnologia que liberta

Edição Especial

24 a 27 de Junho de 2009  
Centro de Eventos PUCRS  
Porto Alegre/RS - Brasil

Maiores informações em:

[www.fisl.org.br](http://www.fisl.org.br)



Promoção



Organização



PROJETO  
SOFTWARE  
LIVRE - RS

PSL-RN  
Projeto Software Livre do  
Rio Grande do Norte

Realização



Transmissão

