

Software Livre para Empresas

Antunes Dantas da Silva^{*}, Gabriel Silva Vinha, Italo M. de Lima Poroca e Valter V. M. de Lucena

^{*}Correspondência:
antunes.dantas@ccc.ufcg.edu.br
Universidade Federal de Campina
Grande, Centro de Engenharia
Elétrica e Informática,
Departamento de Sistemas e
Computação, Rua Aprígio Veloso,
882, Bairro Universitário,
58429-140, Campina Grande,
Brasil.

Resumo

Este trabalho tem como objetivo fazer um estudo sobre o *software* livre e suas aplicações comerciais. Atualmente, o uso de *software* livre tem crescido bastante e é preciso entender o que seria, detalhadamente, *software* livre. Além disso, este trabalho mostra como empresas mudaram seu posicionamento de mercado para se adaptarem a essa realidade e quais são os principais meios de gerar receita através do *software* livre.

Keywords: *software* livre; computação; código aberto; negócios em computação

1 Introdução

O *software* livre é uma realidade que existe desde os primórdios da computação. Baseando-se na ideia básica de que o código fonte deve ser público, o movimento do *software* livre gerou, e ainda gera, bastante polêmica dentre a comunidade da tecnologia da informação, especialmente quando o assunto tange as grandes corporações que lucram com a venda de *softwares* proprietários. Como movimento, iniciou em 1983 com um americano chamado Richard Stallman, que liderou o desenvolvimento de um sistema operacional baseado totalmente nas ideias do *software* livre.

Para ser considerado livre, um *software* deve seguir determinadas "leis", que definem como ele deve ser publicado. Para facilitar a publicação, foram criadas licenças genéricas que servem para qualquer *software*.

Um dos principais questionamentos quando o assunto é tratado é como empresas podem faturar fabricando código aberto. Como será exposto posteriormente, existem diversos modelos de negócios que podem ser abordados para este fim.

Este artigo seguirá a seguinte estrutura: na seção 2, será mostrada a motivação para este estudo. Na seção 3, o tema *software* livre será abordado de maneira mais detalhada, bem como modalidades que onde este é encontrado. Na seção 4, será realizado um breve estudo sobre as principais licenças de publicação. Finalmente, na seção 5, será tratado como empresas podem fazer o uso de *software* livre: tanto no lado cliente quanto no lado empresas produtoras. A seção 6 fará uma discussão sobre o futuro da distribuição dos *softwares* e como o *software* livre se encaixa nessa realidade futura.

2 Motivação

If we assume that sensitive cells follow a deterministic decay $Z_0(t) = xe^{\lambda_0 t}$ and approximate their extinction time as $T_x \approx \frac{1}{\lambda_0} \log x$, then we can heuristically estimate the expected value as:

$$E[Z_1(vT_x)] = \frac{\mu}{r} \log x \int_0^1 x^{1-u} du \quad (1)$$

$$E[Z_1(vT_x)] = \frac{\mu}{r} x^{1-\lambda_1/\lambda_0 v} \log \quad (2)$$

$$1 = 10 \quad (3)$$

$$E[Z_1(vT_x)] = \frac{\mu}{r} \log x \int_0^1 x^{1-u} du \quad E[Z_1(vT_x)] = \frac{\mu}{r} x^{1-\lambda_1/\lambda_0 v} \log \quad (4)$$

Thus we observe that this expected value is finite for all $v > 0$ (also see [1]).

2.1 Exemplo de Sub-Seção

In this section we examine the growth rate of the mean of Z_0 , Z_1 and Z_2 . In addition, we examine a common modeling assumption and note the importance of considering the tails of the extinction time T_x in studies of escape dynamics. We will first consider the expected resistant population at vT_x for some $v > 0$, (and temporarily assume $\alpha = 0$).

$$E[Z_1(vT_x)] = \mu T_x \int_0^{\inf} \lambda_1 T_x(v-u) du \quad (5)$$

If we assume that sensitive cells follow a deterministic decay $Z_0(t) = xe^{\lambda_0 t}$ and approximate their extinction time as $T_x \approx -\frac{1}{\lambda_0} \log x$, then we can heuristically estimate the expected value as.

3 Software Livre

Table 3 shows the average α and the standard deviation for the CCR [1] obtained by the *GLCM+SOM* method. We can conclude that for the Brodatz dataset [2] the processing tool based on mean vectors is the best option [1, 3]. Considering this result [4], the mean vector approach is adopted as processing tool of the *GLCM+SOM* method for the next experiments [5].

testando 123

Tabela 1 - Sample table title. This is where the description of the table should go.

	B1	B2	B3
A1	0.1	0.2	0.3
A2
A3

3.1 Software as a Product

aqui vc faz

3.2 Software as a Service
olar

3.3 Componentes da Produção de Software
acesso ao software

4 Licenças de Publicação

tarara

5 Software Livre Para Empresas

O uso de *software* livre em ambientes domésticos e comerciais tem crescido bastante nos últimos anos. O governo brasileiro, inclusive, tem investido bastante em soluções livres nos seus computadores. O mercado em geral também está abraçando a ideia e isso tem feito com que cada vez mais empresas surjam com foco no desenvolvimento de aplicações baseadas em *software* livre.

Para o cliente, há inúmeros benefícios no uso de *software* livre em suas máquinas. Em primeiro lugar: economia. Para o usuário final, isso significa menos um custo. Para empresas, uma enorme economia.

Para facilitar, pode-se imaginar uma empresa imaginária de *call center*: em uma empresa deste tipo, há muitos computadores. A empresa imaginária teria 1000 computadores para uso dos atendentes mais 200 computadores para uso de supervisores e gerentes. No total 1.200 máquinas que precisam de um sistema operacional e, no caso das máquinas dos supervisores e gerentes, suíte de escritório.

Levando em conta que o custo médio da licença de um sistema operacional proprietário é R\$ 200,00, apenas com este recurso, a empresa gastaria R\$ 240.000,00. Com o uso de *software* livre, a empresa economizaria bastante dinheiro ao adotar um sistema operacional de código aberto, como o Debian.

Além do benefício financeiro, *softwares* livres são, geralmente, mais seguros. Isso acontece por que com o código fonte disponível, qualquer programador pode descobrir um *bug* (erro) e submeter uma correção para ele. O Linux, *kernel* de sistema operacional livre mais difundido no mundo é considerado o sistema operacional mais seguro.

Outro benefício é a facilidade na alteração do funcionamento de algum programa. Com o código aberto, um empresário, por exemplo, pode requisitar a alguma empresa que altere o comportamento de um programa, ou acrescente algo, para adequá-lo à sua realidade e aos seus problemas.

Porém, para o lado das fabricantes de *software*, quais as vantagens de fabricar *software* livre?

Muitas pessoas quando ouvem o termo *software* livre o associam à *software* gratuito. Isso não é verdade. *Software* livre não quer dizer *software* gratuito e existem diversas abordagens comerciais que podem ser utilizadas para monetizar programas de código aberto.

Existem inúmeras empresas que lucram com o *software* livre. A Canonical, desenvolvedora da distribuição Linux Ubuntu tem uma receita anual de \$ 65,7 milhões de dólares. Ela se mantém através de doações e acordos comerciais para inserir conteúdo patrocinado dentro dos seus *softwares*.

A Mozilla, empresa responsável pelo popular navegador de internet Firefox tem uma receita de \$ 330 milhões de dólares anuais. O Firefox, mesmo sendo um *softwares* livre, gera boa parte dessa receita através de acordo com empresas para agregação de serviços (como o Google Search ser o buscador padrão do navegador) e propagandas em algumas seções do navegador, como a aba "Nova Guia".

A Red Hat é, sem dúvida, uma das mais bem sucedidas empresas no ramo do *softwares* livre. Ela atua através do fornecimento de soluções corporativas. Seus principais produtos são armazenamento, sistemas operacionais, consultoria, treinamento e suporte. Ela utiliza várias abordagens de mercado para vender seus serviços e isso tem dado muito certo. Em 2014, sua receita foi de \$ 1,53 bilhões de dólares. Seus serviços são amplamente utilizados e bem aceitos no mercado.

Essas empresas todas trabalham desenvolvendo *software* livre e tem uma receita considerável. Existem inúmeros outros exemplos de sucesso com o *software* livre: Android, Apache, LibreOffice, Swift são exemplos de aplicações de sucesso no mundo do *software* livre.

Existem inúmeras abordagens de mercado para fatura com *software* livre. Venda de suporte, treinamento, consultoria, propaganda, dentre outros, são alguns exemplos disso. A seguir, será apresentado as três principais abordagens de mercado para o *software* livre.

5.1 estatísticas de mercado para saap

oi

5.2 Software as a service

aqui vc faz

5.3 Open Core

O Open Core é um modelo de negócio baseado no uso misto entre *software* livre e *software* fechado. No Open Core, as empresas disponibilizam uma versão básica, porém útil, de algum sistema feito como *software* livre e vendem complementos para esse sistema. Esse complementos, como o próprio nome deixa claro, acrescenta alguma funcionalidade no sistema. Existem inúmeras soluções baseadas no Open Core, como o MySQL da Oracle. É um modelo de negócio muito interessante pois permite que empresas foquem no desenvolvimento de partes pequenas do sistema, produzindo com melhor qualidade e podendo oferecer um melhor suporte. A já citada Red Hat também faz uso do Open Core no seu serviço JBoss.

Para os clientes, é uma abordagem bastante interessante por que permite economizar ao usar um grande *software* livre gratuito e comprar apenas complementos que esta precise.

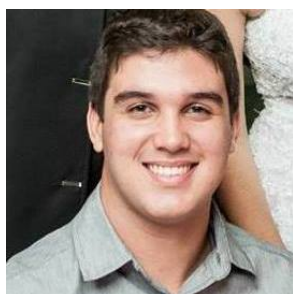
Há controvérsias acerca da legitimidade do Open Core como sendo *software* livre. Muitas pessoas dizem que pelo fato dos complementos serem, geralmente, de código fechado, não há como classificá-lo como *software* livre. Porém, é uma estratégia de bom senso e que vem dando certo no mercado em geral.

6 Tendências

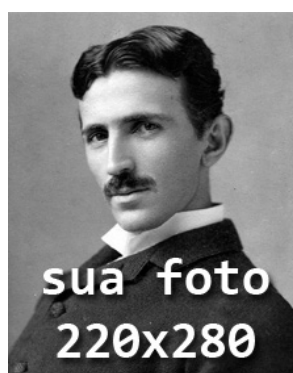
olar

Referências

1. Rosenfeld, A., Troy, E.B.: Visual Texture Analysis. In: A Symposium on Feature Extraction and Selection in Pattern Recognition, vol. 1, pp. 115–124 (1970)
2. de Oliveira Domingues, M.A.: Métodos robustos de regressão linear para dados simbólicos do tipo intervalo. PhD thesis, Universidade Federal de Pernambuco (UFPE), Recife (2010)
3. Diday, E.: Introduction à l'analyse des données symboliques. Research report, Institut National de Recherche en Informatique et en Automatique, INRIA (1989)
4. Lutkebohle, I.: BWorld Robot Control Software. <http://www.nlm.nih.gov/research/visible/> - accessed 19-July-2008
5. de tal, F.: Livro de Eng. Civil, 6th edn. McGraw-Hill, Inc., New York, NY, USA (2005)



Antunes Dantas da Silva Nasceu em Picuí, interior do estado brasileiro Paraíba. Possui formação técnica em Manutenção e Suporte em Informática pelo Instituto Federal da Paraíba Campus Campina Grande. Foi membro do Grupo de Pesquisa em Redes Convergentes. Atualmente, está na graduação em Ciência da Computação, pela Universidade Federal de Campina Grande. É membro do Programa de Educação Tutorial, onde desenvolve o Sistema de Avaliação Docente, orientado pelo professor doutor Matheus Gaudêncio do Rêgo. Tem interesse em Engenharia de Software, Desenvolvimento de Sistemas e Telecomunicações.



Fulano de Tal was born in India. She received the B.S. degree in computer science from Kurukshetra University, Kurukshetra, India and the M.Phil. and Ph.D. degrees from the University of Exeter, Exeter, UK in 1999, 2001 and 2004, respectively. Her Ph.D. was in the area of machine learning for image analysis in aviation security. Her main research interests include image processing, natural scene analysis, video analysis, and neural networks. She has published more than 30 papers in the area of machine learning for image analysis in peer reviewed journals and conferences. Currently she is a Senior Research Fellow at Loughborough University leading the project on imaging for road transport applications.