ANÁLISE DOS SOFTWARES MAIS UTILIZADOS EM UM CURSO DE ENGENHARIA DE COMPUTAÇÃO

Kléber Macedo Cabral – klebermc1@gmail.com
Rafael Sobrinho Mendanha – rafaelsmendanha@gmail.com
Adriano César Santana – adriano@emc.ufg.br
Karina Rocha Gomes da Silva – karinarg@emc.ufg.br
Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação
Av. Universitária, n. 1488, Quadra 86, Bloco A, 3º piso – Setor Leste Universitário
74605-010 – Goiânia – Goiás

Resumo: *Este artigo tem como objetivo primário analisar os principais softwares utilizados no curso de Engenharia de Computação da Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) na Universidade Federal de Goiás (UFG). Realizou-se um mapeamento dos softwares utilizados ao longo do curso, tendo como método duas fontes de informações: a) Análise do Projeto Político Pedagógico do Curso (PPP); b) Entrevista com os professores que utilizam algum tipo de software em suas disciplinas. Após discorrer brevemente sobre as possibilidades de classificação de softwares, os mesmos foram agrupados em: código aberto e software proprietário. Com as análises e a classificação realizadas, foi possível identificar os quatro softwares mais utilizados, e a partir disso, detalhar seus pontos fortes, fracos e suas características. Por fim, esse estudo mostra que é importante considerar possíveis alternativas de softwares de Código Aberto como uma forma de ampliar a acessibilidade dos alunos e economia de gastos financeiros para a Universidade.*

Palavras-chave: *Software livre, Código aberto, Código proprietário, Engenharia.*

1. INTRODUÇÃO

Durante o curso de Engenharia de Computação, o aluno tem contato com diversas ferramentas de software que são auxiliares ao ensino. Essas são utilizadas com o intuito de simplificar os cálculos e as modelagens que são por vezes trabalhosos, sendo constantemente exigidas durante o período de aprendizagem (AGOSTINHO et al., 2006).

A importância do uso de softwares provém do fato que esses auxiliam, entre outras coisas, o aprendizado mais efetivo do conteúdo estudado, fazendo com que o aluno pratique os conhecimentos teóricos aprendidos em sala, além de aproximar o aluno do ambiente profissional.

Visto que, os softwares têm um valor tão alto durante o processo de aprendizagem, algumas questões foram chave para o desenvolvimento do trabalho, tais como: Esses softwares estão disponíveis aos alunos? Qual a importância desses softwares nas disciplinas teóricas? E nas práticas? O que mudaria se a EMC/UFG escolhesse por alternativas livre / código aberto?

Tendo como motivação a necessidade de um estudo sobre o uso de softwares durante o período de graduação em Engenharia de Computação na Universidade Federal de Goiás (UFG), apresenta-se nesse trabalho os dados obtidos por meio do levantamento realizado e posteriormente é feita a análise destes. Uma relação contendo os softwares e suas características auxiliam, entre outros, a coordenação do curso ao se definir estratégias e objetivos de ensino.

Sabe-se que nos dias de hoje existem diversas classificações para as ferramentas de software, sendo assim convém separá-las em grupos menores de acordo com suas características em comum. Nesse artigo explica-se melhor como são essas divisões e como foram classificados os softwares obtidos na pesquisa.

As seções seguintes serão organizadas da seguinte forma: A Seção 2 discorre sobre como as licenças de software podem ser caracterizadas, de acordo com suas políticas de uso e distribuição. A Seção 3 apresenta o projeto pedagógico do curso e qual foi a metodologia de pesquisa utilizada. A seção 4 mostra os resultados obtidos, ou seja, a relação de softwares por disciplina e as considerações feitas. A seção 5 prossegue explicando sobre os quatro softwares que são utilizados em um número maior de disciplinas, de acordo com a pesquisa. A seção 6 apresenta uma análise dos resultados obtidos e discute sobre as questões levantadas pela pesquisa. Por fim, a seção 7 traz as considerações finais do trabalho.

2. CARACTERIZAÇÃO DE SOFTWARES DE ACORDO COM SUA LICENÇA

A seguir, apresentam-se os tipos de softwares e como foram classificados de acordo com sua licença. São explicitadas as diferenças existentes entre eles e como podem ser colocados em categorias distintas de acordo com a forma que serão utilizados, distribuídos ou modificados.

2.1. Definição de Software Livre

De acordo com a Free Software Foundation (FSF, 2011), um software será livre quando respeitar os seguintes princípios de liberdade:

- *A liberdade de executar o programa para qualquer propósito:* O que significa execução sem restrições impostas pelo fornecedor, da forma como o usuário julgar melhor.
- *A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades:* Isso implica no acesso ao código-fonte, na estrutura do software, e sua possível alteração.
- *A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo:* A essa liberdade é que estão atreladas as maiores discussões, pois ela exige a distribuição de forma gratuita e sem restrições.
- *A liberdade de aperfeiçoar o programa, e divulgar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie:* Aqui entra o caso de distribuição de cópia alterada, com o mesmo nome da original. Como isso pode, em determinadas circunstância, ferir a credibilidade do fornecedor, também é um forte motivo para a não adequação as a política de software livre (GUGIK, 2011).

2.2. Definição de Software de Código Aberto

Outra caracterização de software, que foi proposta pela Open Source Initiative (OSI, 2011), é a que denomina os chamados softwares de Código Aberto (Open Source) e que está disponível no site da fundação. Apesar de conter mais regras do que os quatro princípios propostos pela Free Software Foundation, pode-se afirmar que é uma definição derivada da primeira, só que mais abrangente e relacionada com a parte técnica do software, ao invés de conceitos éticos como é a proposta pela FSF (SABINO & KON, 2009).

A definição de Código Aberto concede ao desenvolvedor o direito de estabelecer algumas restrições, como por exemplo, a proibição da distribuição de uma versão alterada do código com o mesmo nome do original (GUGIK, 2011). Neste artigo, utiliza-se o termo Código Aberto para também identificar as ferramentas que são Software Livre, visto que de modo geral, as licenças que atendem à Definição de Software Livre (da Free Software Foundation) também atendem à Definição de Código Aberto (da Open Source Initiative), e assim pode-se dizer (na ampla maioria dos casos) que se um determinado software é livre, ele também é de código aberto, devido as características das duas classificações.

2.3. Definição de Software Proprietário

De um modo grosseiro, pode-se classificar como software proprietário todo aquele que não se enquadra, de alguma forma, nas definições de software livre e de código aberto, fornecidas pela Free Software Foundation e pela Open Source Initiative respectivamente (GNU, 2011). Usualmente o software proprietário tem restrições sobre reprodução, distribuição ou uso do mesmo. No artigo é considerado como software proprietário aquele que, sendo gratuito ou não, entre outras características, não nos permite acesso ao seu código fonte.

3. PROJETO PEDAGÓGICO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

Com as definições de software anteriormente apresentadas e tomando por base o projeto pedagógico do curso, aprovado em 2008, de onde foram retirados os nomes das disciplinas do curso junto com os institutos responsáveis por oferecê-las (PPP, 2011), foi realizado um estudo sobre a utilização de ferramentas de código aberto e proprietário no curso de Engenharia de Computação da UFG.

A escolha do software utilizado em cada disciplina é, de maneira geral, a critério do docente, sendo assim optou-se por fazer a pesquisa em campo consultando apenas os professores das disciplinas da parte de engenharia do curso, que são concedidas pela EMC. Como a EMC é a instituição que administra o curso, a relação entre professor - disciplina lecionada é maior, pois os demais institutos apenas decidem qual o docente ministrará a disciplina quando há a demanda da mesma. Sendo assim, a cada ocorrência de uma mesma disciplina ao longo dos anos, a probabilidade de que seja o mesmo docente (e consequentemente o mesmo software) é maior nas disciplinas da parte de engenharia do curso.

Devido a essa escolha, disciplinas da área de computação, entre elas algoritmos, estruturas de dados e compiladores não foram tratadas por serem de responsabilidade de outro instituto, o Instituto de Informática (INF). Essas disciplinas estão listadas no Projeto Político Pedagógico do curso (PPP, 2011).

3.1. Metodologia de pesquisa

Para as disciplinas se enquadram na descrição acima, realizou-se a coleta de dados sobre cada uma, junto aos professores da EMC, para obter os nomes dos softwares utilizados nas naquelas em que usualmente lecionam, ou que lecionaram em um tempo próximo. A abordagem foi feita oralmente e por e-mail.

4. APRESENTAÇÃO DOS RESULTADOS

Os resultados da pesquisa de campo estão dispostos na Tabela 1. A seguir, apresentam-se algumas considerações tomadas, junto com as suas justificativas, para chegar a esses resultados.

Estão na Tabela 1 apenas as disciplinas que utilizam algum tipo de software específico. Além disso, as disciplinas de Tópicos e dos grupos específicos presentes no projeto pedagógico (PPP, 2011), não foram consideradas na pesquisa. Isso se justifica porque as ementas são decididas pelos docentes no momento em que estas são ofertadas. Sendo assim não existe um software específico nestes casos.

Também estão agrupadas as disciplinas que possuem parte prática e teórica separadas, como por exemplo, Circuitos Elétricos e Laboratório de Circuitos Elétricos. Essa simplificação é feita apenas por praticidade na apresentação dos resultados, e justificada pelo fato que os softwares utilizados tendem a serem os mesmos. Embora os resultados estejam assim, ainda são consideradas disciplinas distintas, ou seja, na análise que será feita adiante serão consideradas duas disciplinas ao invés de uma, por exemplo.

Tabela 1 – Softwares utilizados em cada disciplina.

Disciplina	Software Proprietário	Software de Código Aberto
Análise de Sistemas Lineares	MATLAB.	
Antenas e Propagação	Pathloss.	Rpath, Dpath, Dpath2.
Arquitetura de Computadores 1	MARS.	Icaros Verilog, Gtkwave.
Arquitetura de Computadores 2	MARS.	-
Automação de Processos Contínuos Industriais	Labview, MATLAB.	-
Automação de Processos Discretos Industriais	Labview, MATLAB.	-
Circuitos Elétricos 1*	PSPICE/ORCAD, PSIM, PSCAD.	-
Circuitos Elétricos 2	PSPICE/ORCAD, PSIM, PSCAD, Multisim.	-
Circuitos Lógicos*	PSPICE/ORCAD.	Oregano, NGSpice.
Comunicações Digitais	MATLAB.	-
Controle Digital	Labview, MATLAB.	-
Controle Moderno	MATLAB.	-
Conversão Eletromecânica de Energia 1*	MATLAB, PSCAD.	-
Desenho	Auto Cad.	-

Eletrônica 1*	PSPICE/ORCAD, PSIM, Proteus.	-
Eletrônica 2	PSPICE/ORCAD, PSIM, PSCAD, Multisim.	-
Eletrônica Digital	PSPICE/ORCAD, Altera.	-
Engenharia Econômica	EXCEL.	-
Instrumentação, Sensores e Atuadores	Labview, MATLAB.	-
Disciplina	**Software Proprietário**	**Software de Código Aberto**
Materiais Elétricos para Engenharia de Computação*	PSPICE/ORCAD.	-
Mecânica e Resistência dos Materiais	Analysis, Ftool.	-
Microprocessadores e Microcontroladores*	Proteus.	Pequi.
Modelagem Matemática de Sistemas Dinâmicos.	Labview, MATLAB.	-
Pesquisa Operacional 1	MATLAB.	C, Java.
Pesquisa Operacional 2	MATLAB.	-
Projetos com Microcontroladores	Proteus, MPLAB, Hightec**, KEYL**.	-
Robótica 1	Labview, MATLAB.	-
Robótica 2	Labview, MATLAB.	-
Sistemas de Comunicações	MATLAB.	-
Sistemas de Comunicações Móveis	MATLAB.	-
Sistemas de Controle*	MATLAB.	-
Teoria de Telecomunicações	MATLAB.	-

* Disciplinas de laboratório e teórica agrupadas.
** Descrevem uma família de software dessas empresas.

5. SOFTWARES MAIS UTILIZADOS

Ao analisar o número de disciplinas em que cada software foi listado, obtém-se então os quatro que aparecem como mais utilizados. Segue uma breve descrição sobre cada dos quatro, eles são: MATLAB, OrCAD PSpice, LabVIEW e PSIM. Observa-se que pelo menos um deles é utilizado em 77% das disciplinas. Na Figura 1 é apresentado o valor percentual das disciplinas que utilizam cada um deles.

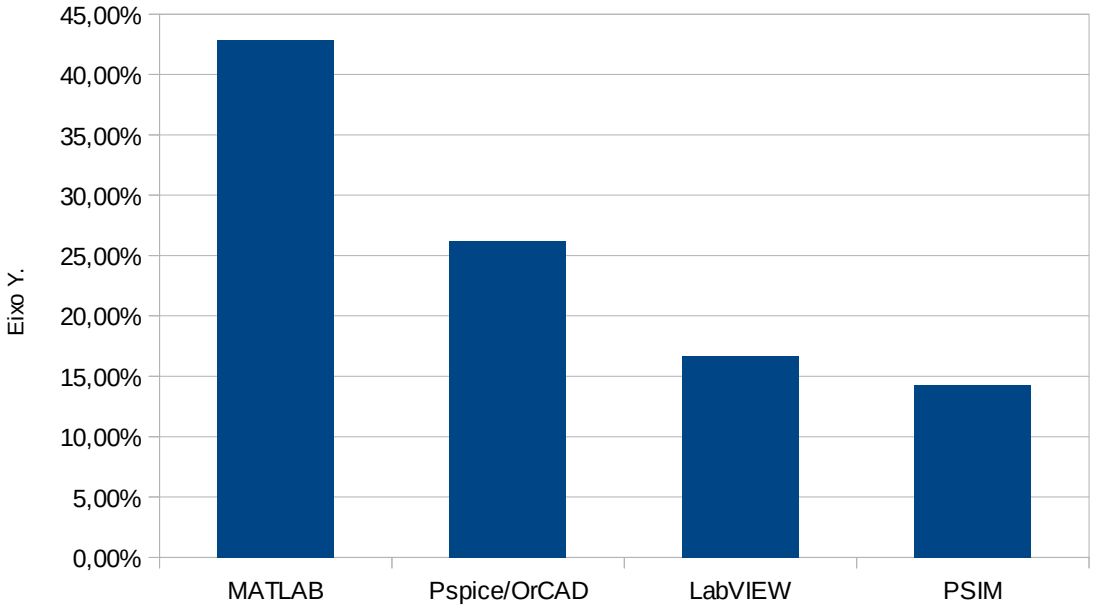

Figura 1 - Os quatro softwares mais utilizados dentre as disciplinas consideradas. No eixo Y tem-se o valor percentual das disciplinas que utilizam cada um dos softwares.

Segue a descrição sobre cada software apresentado na figura 1:

- MATLAB é a abreviação de MATrix LABoratory. Possui um ambiente interativo permitindo a computação numérica, análise de dados e sua visualização e desenvolvimento de algoritmos. Podem ser adicionados pacotes de funções chamadas toolboxes. Estas adicionam ao MATLAB funções específicas relacionadas a diversas áreas, como: processamento de sinais, otimização, estatística, sistemas de controle entre outros (MATHWORKS, 2011);
- OrCAD Pspice possibilita o projeto e a simulação de circuitos que trabalham com sinais mistos e analógicos (CADENCE, 2011). É fundamental em disciplinas que abordam circuitos elétricos, eletrônica e materiais elétricos;
- LabVIEW é um ambiente gráfico de programação, medições e controle de sistemas muito utilizado por cientistas e engenheiros (NATIONAL INSTRUMENTS, 2011). Permite a integração com diversos dispositivos de hardware, como sensores, atuadores entre outros. Contém bibliotecas que possibilitam a análise e visualização de dados, podendo apresentá-los também em três dimensões;
- PSIM permite através de uma interface amigável a criação e simulação de circuitos elétricos. Foi desenvolvido especialmente para circuitos de eletrônica de potência e controle de motores (POWERSIM, 2011). Pode ser utilizado tanto em circuitos digitais quanto em circuitos analógicos.

6. ANÁLISE E DISCUSSÃO

Esta seção inicia apresentando uma análise do percentual das disciplinas que utilizam apenas software proprietário, apenas software livre ou ambos. Prossegue fazendo algumas

considerações para as questões que foram apresentadas na seção de introdução e que serviram como direção para o trabalho, tomando como base a visão dos alunos.

6.1. Os tipos de software e sua relação com as disciplinas

Observando os resultados da Tabela 1, verifica-se que nenhuma dessas disciplinas utiliza apenas software de Código Aberto, trinta e duas (84,21%) utilizam apenas software proprietário e apenas sete (15,79%) utilizam ambos os tipos. A figura 2 representa graficamente essas informações. Nele, *Proprietários* representa o número de disciplinas em que se utiliza apenas software proprietário, *Código aberto* mostra o número de disciplinas que utilizam apenas esse tipo de software e *Código aberto e proprietário* representa o número de disciplinas que utiliza os dois tipos de softwares.

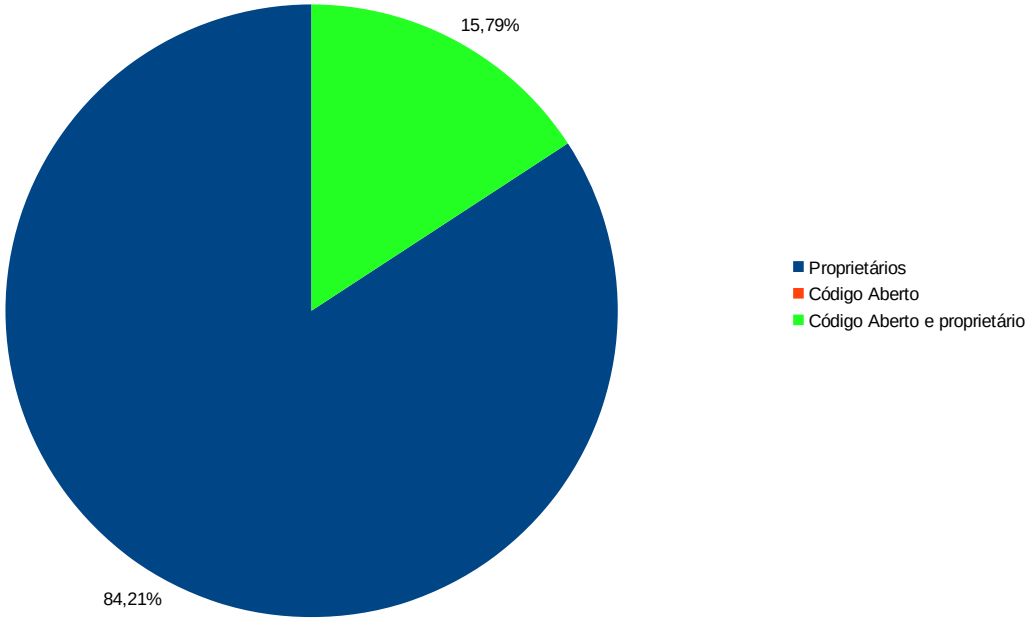

Figura 2 – Valor percentual das disciplinas que utilizam software livre, proprietário ou ambos.

6.2. Acessibilidade dos softwares

- Esses softwares estão disponíveis aos alunos de forma acessível?

Para os softwares identificados como mais utilizados, todos são proprietários e a maioria tem licença não gratuita, assim sendo, os alunos tem acesso aos mesmos por meio da aquisição das licenças, dos laboratórios da instituição de ensino, ou através de formas ilegais na internet. A primeira opção é muitas vezes inviável, dado o valor cobrado, a segunda pode não ser possível, dado o número de computadores ou número de licenças disponíveis nos laboratórios da instituição, sendo a terceira alternativa a mais usual.

Sabe-se que existem os softwares que mesmo sendo proprietários, tem licença gratuita (em inglês chamados de freeware). Apesar de terem sim a vantagem da acessibilidade, sendo geralmente fácil de obtê-los pela internet, os mesmos ainda esbarram nas restrições de uso, impostas pela sua licença, e no baixo (ou nenhum) acesso ao código fonte, não podendo ser estudadas ou melhoradas.

6.3. E quanto à utilização de código aberto?

- O que mudaria se a instituição de ensino escolhesse por alternativas livre / código aberto?

Sabe-se que as ferramentas caracterizadas como código aberto ou software livre têm algumas vantagens didáticas em relação às proprietárias. Utilizando ferramentas de código aberto e sendo devidamente orientados, os alunos podem ampliar os horizontes de aprendizado, pois não estão limitados a apenas utilizar as ferramentas, podendo também estudá-las a fundo, suas estruturas e seus componentes, e porventura propor melhorias. Não esquecendo a acessibilidade proveniente de sua distribuição, que é por vezes gratuita.

6.4. Importância dos softwares no aprendizado

- Qual a importância dos softwares nas disciplinas teóricas? E nas práticas?

Ao utilizar as ferramentas de software o aluno consegue colocar em prática o seu conhecimento. Em disciplinas puramente teóricas, o software enriquece o aprendizado, pois permite o exercício dos conceitos vistos em sala. Em disciplinas com parte prática é quase imprescindível, pois o uso do software geralmente viabiliza a simulação das experiências realizadas e a agilidade nos cálculos necessários em ambiente de laboratório.

7. CONSIDERAÇÕES FINAIS

Neste trabalho inicialmente são definidos os tipos de softwares e como se dá sua classificação. Não entrando em pormenores dos princípios éticos das definições possíveis, se considera uma classificação relativamente ampla. Utilizam-se ao longo do texto as definições de Código Aberto e de Software Proprietário.

Através do Projeto Pedagógico do curso de Engenharia de Computação da UFG, foram coletadas e caracterizadas as ferramentas de software que são utilizadas durante o período de graduação, ou seja, quais os softwares utilizados nas disciplinas do curso, especificamente sobre as disciplinas da área de engenharia. Apresentaram-se os motivos que levaram apenas ao estudo das disciplinas ofertadas pela EMC. Após a coleta das informações, viu-se que nenhuma das disciplinas utilizava apenas Softwares Livres e que apenas 17% utilizavam ambos os tipos de software. Segue um trecho onde se apresenta os quatro softwares mais utilizados, sendo eles: MATLAB, OrCAD Pspice, LabVIEW e PSIM.

Analisando essas informações, realizou-se uma importante discussão sobre fatores como acessibilidade das ferramentas de software utilizadas. Por meio da opinião de alunos verifica-se que devido à alta utilização de softwares proprietários, a instituição, docentes e discentes ficam presos à necessidade da aquisição das licenças, que muitas vezes não são gratuitas, e/ou ao uso restrito das ferramentas, imposto pelo seu fornecedor. Vê-se que seria mais proveitoso o uso de softwares código aberto pela facilidade de sua distribuição, que por vezes gratuita, além do incentivo a exploração do seu funcionamento e componentes internos por parte dos alunos.

Com esse estudo, considera-se importante uma análise futura mais detalhada sobre possíveis alternativas de Código Aberto para os softwares proprietários que foram apresentados na Tabela 1, como uma forma de ampliar a acessibilidade dos alunos aos softwares e de economia de recursos para a instituição.

REFERÊNCIAS BIBLIOGRÁFICAS

AGOSTINHO, N. U.; NEVES, L. D. R.; BARBOSA, A. M.; NEPOMUCENO, E. G.; Softwares livres no ensino de engenharia: uma atitude socialmente justa, economicamente viável e tecnologicamente sustentável. Anais: XXXIV Congresso Brasileiro de Ensino de Engenharia. Passo Fundo: Universidade de Passo Fundo, 2006.

CADENCE. **PSpice A/D and Advanced Analysis**. Disponível em: <http://www.cadence.com/products/orcad/pspice_simulation/pages/default.aspx> Acesso em: 16 nov. 2011.

CAMPOS, A. **O que é software livre.** Disponível em: <http://br-linux/faq-softwarelivre/> Acesso em: 19 out. 2011.

FSF. **Free Software Foundation.** Disponível em: <http://www.fsf.org/>. Acesso em: 19 out. 2011.

GNU. **Categories of free and nonfree software.** Disponível em: <http://www.gnu.org/philosophy/categories.html>. Acesso em: 17 nov. 2011.

GUGIK, G. **Código aberto e software livre não significam a mesma coisa.** Disponível em: <http://ww.tecmundo.com.br/1739-codigo-aberto-e-software-livre-nao-significam-a-mesma-coisa-.htm> Acesso em: 19 out. 2011.

MATHWORKS. **Matlab: Product Description.** Disponível em <http://www.mathworks.com/products/matlab/description1.html>. Acesso em: 20 out. 2011.

MCIA. **Simulation Lab.** Disponível em: <https://genweb.upc.edu/mcia/facilities/simulation-lab> Acesso em: 16 nov. 2011.

NATIONAL INSTRUMENTS. **Product Information: What is NI LabVIEW?.** Disponível em <http://www.ni.com/labview/whatis/> Acesso em: 16 nov. 2011.

OSI. **Open Source Initiative.** Disponível em: <http://www.opensource.org/>. Acesso em: 19 out. 2011.

POWERSIM. **PSIM.** Disponível em: <http://www.powersimtech.com/index.php?name=psim> Acesso em: 16 nov. 2011.

PPP. **Projeto Político Pedagógico Curso de Engenharia de Computação.** Disponível em: <http://www.eeec.ufg.br/?menu_id=9&pos=esq&site_id=1>. Acesso em: 17 nov. 2011.

SABER ELETRÔNICA. **Minicurso de LabVIEW Aprenda fazendo.** Disponível em: <http://www.sabereletronica.com.br/secoes/leitura/1643> Acesso em: 16 nov. 2011.

SABINO, Vanessa; KON, Fábio. UNIVERSIDADE DE SÃO PAULO, Instituto de Matemática e Estatística, Centro de Competência em Software Livre. Licenças de Software Livre: História e características, 2009. 36p, Relatório Técnico.

ANALYSIS OF COMMONLY USED SOFTWARE IN A COURSE OF COMPUTER ENGINEERING

Abstract: *This article have as primary objective to analyze the main softwares used along the Computer Engineering course of School of Electrical, Mechanics and Computer Engineering (EMC) at the Federal University of Goiás (UFG). First was performed a mapping of softwares used along the course, having as the input method two sources of information: a) Analysis of the Political Pedagogical Project of the Course (PPP); b) Interview with the teachers who use software in their classes. After a short discourse about the classification possibilities for softwares, they were classified in two groups: open source and proprietary software. Once the analysis and the classification were done, was able to identify what were the four most used softwares, pointing its strengths, weaknesses and its features. Finally, this study shows that is important to consider alternatives open source softwares as a way to extend the accessibility of the students and to save financial costs to the university.*

Keywords: *free software, open source, proprietary, engineering.*