

ESTUDO DE CAD LIVRE PARA IMPLEMENTAÇÃO DE FERRAMENTA DE PROJETO

Free, shared and open cad study for the development of a design tool

Sergio Adriano Bizelloⁱ e-MAIL | CV Lattes

Regina Coeli Ruschelⁱⁱ e-MAIL | CV Lattes

Resumo

O setor de projetos de Arquitetura Engenharia e Construção (AEC) têm sua principal ferramenta de desenvolvimento sistemas de projeto auxiliado por computador (sistemas CAD), baseado em software proprietários, com custos elevados e necessários em várias estações de trabalho. Entretanto, atualmente os usuários de sistemas CAD no Brasil e no mundo carecem de um programa do tipo software livre ou open source que possa ser equiparado ao software proprietário padrão e líder de mercado, o AutoCAD®, da Autodesk, viabilizando redução de custos com Tecnologia da Informação. O estudo apresentado procura um sistema CAD alternativo que possa abrigar a metodologia de projeto automatizada. Esta pesquisa caracteriza-se como descritiva de sistemas CAD livre sendo desenvolvida nas seguintes etapas: levantamento, categorização, triagem, definição do sistema CAD de controle, definição de elementos e critérios de avaliação, aplicação da avaliação e análise. Os sistemas avaliados são apresentados e comentados. É possível dizer que ainda não há um sistema CAD livre que possa abrigar o desenvolvimento de uma ferramenta de projeto com características conjuntas de recursos em 2D, 3D e extensibilidade. Como contribuição apresenta-se um roteiro para escolha de um sistema CAD alternativo ao software proprietário para o desenvolvimento de projeto de AEC.

Palavras-chave: CAD, software livre, open source, modelagem, projeto arquitetônico.

Abstract

Development tools in Architecture Engineering and Construction (AEC) are based on high costs computer-aided design (CAD) proprietary software installed on multiple workstations. However, nowadays the CAD systems, Brazilian or worldwide, need a free software program comparable to AutoCAD, standard proprietary software and market leader, in order allow Information Technology cost reduction. This study is motivated by the search of an alternative CAD system in order to implement a housing automated design methodology. This research is characterized as descriptive of free CAD systems developed in the following stages: collecting, categorizing, sorting free CAD systems, defining a proprietary CAD system of control comparison, defining elements and assessment criteria, evaluation and analysis. The free CAD systems evaluated are presented and discussed. It was observed that still there isn't a free CAD system which can accommodate the development of a tool for automated design methodology including simultaneously features of 2D, 3D and extensibility. As a contribution it is presented guidelines for choosing a CAD system alternative to proprietary software for the AEC design.

Keywords: CAD, free software, open source, modeling, architectural design



1. INTRODUÇÃO

Nos dias de hoje é difícil encontrar um escritório de projeto que utilize a prancheta e demais ferramentas para o desenho técnico tradicional. Alguns profissionais necessitam utilizar suas habilidades manuais para realizar o trabalho de concepção de projeto, porém nas fases posteriores (como de anteprojeto e projeto executivo) as exigências de mercado fazem com que este trabalho seja digitalizado por algum meio (RUSCHEL; FREITAS, 2000). Do anteprojeto em diante o desenho não volta mais a sua forma não eletrônica sendo que esse trabalho, então, é realizado com sistemas CAD. É possível encontrar vários sistemas deste tipo com os quais é possível realizar um projeto de AEC. Além de confeccionar os desenhos necessários do anteprojeto até o detalhamento final do empreendimento, os profissionais podem também realizar a concepção do projeto utilizando sistemas CAD.

Freitas e Ruschel (2000) dizem que quantitativamente o nível de utilização dos programas CAD é alto. Qualitativamente também se torna comum à utilização de recursos como maquetes eletrônicas, apresentações com alto grau de realismo, passeios virtuais e até mesmo modelos de projetos que carregam informações sobre o empreendimento (NASCIMENTO; SANTOS, 2003). Panizza e Ruschel (2005) afirmam que a qualidade de um projeto elaborado num sistema CAD não se restringe à aparência do desenho impresso, mas também nas informações contidas nele. Isso melhora a eficiência do desenvolvimento do projeto num possível ambiente colaborativo. Nesse cenário, entretanto, a maioria dos escritórios de projetos se vale ainda do desenho bidimensional ou 2D para confeccionar suas pranchas de desenho procurando produtividade (TSE; WONG; WONG, 2005).

Assim mesmo, para a elaboração de um desenho bidimensional é necessário um número mínimo de ferramentas essenciais para a elaboração deste tipo de projetos,

visando garantir precisão e qualidade no trabalho. As ferramentas e recursos que um programa do tipo CAD pode apresentar são diversos, e sua utilização ou não pode variar de acordo com o grau de conhecimento do usuário e sua experiência na área. De qualquer modo, existem ferramentas básicas de diversas finalidades sendo que a maneira como é usada pode influenciar até mesmo em produtividade, embora esse trabalho não leve em conta este aspecto. Cada ferramenta executa uma ação diferente, entretanto é possível agrupá-las por funcionalidade, dentro do programa. Algumas das ferramentas essenciais fazem parte também dos conceitos envolvidos na qualidade dos arquivos CAD no que se diz respeito a uma utilização colaborativa que é um uso mais avançado desses sistemas (PANIZZA; RUSCHEL, 2005).

A motivação para a realização deste trabalho advém da necessidade encontrada de flexibilização da ferramenta computacional AUTOMET (RUSCHEL; RODRIGUES, 2002). A ferramenta AUTOMET desenvolve projetos arquitetônicos em 2D e 3D para casas térreas de interesse social, mais especificamente é voltada para a autoconstrução. A ferramenta AUTOMET é atualmente implementada em Autolisp e funciona como um aplicativo dentro do sistema AutoCAD®. Assim é desejável para que a referida ferramenta seja amplamente divulgada, utilizada e continuamente ampliada de forma irrestrita que a mesma opere num sistema operacional e CAD livres. Este trabalho está de acordo com as expectativas de Novaes e Coelho (2005) que sugerem que as instituições de ensino devem fomentar o desenvolvimento de pesquisas com software livre. Verifica-se também a importância e alinhamento desta pesquisa com preocupações do governo federal representada na Lei No. 11.888 que assegura às famílias de baixa renda assistência técnica pública e gratuita para o projeto e a construção de habitação de interesse social (BRASIL, 2008).

2. SOFTWARE LIVRE

Muitas pessoas perguntam por que usar *software* livre. Nos dias de hoje nossa sociedade é extremamente dependente de computadores e um grande exemplo disso foi a fortuna gasta para corrigi-los durante a

virada do milênio (ALMEIDA, 2000). Ainda sob esta ótica deste autor, um instrumento que tem sua importância comparada ao papel e lápis em tempos atrás, como é o computador, não pode depender de que a

maior parte dos programas que comandam o seu funcionamento seja fornecida por um único só fabricante. Por essa razão encontram-se, no mundo todo, movimentações em todos os níveis da sociedade para que esse quadro possa mudar. Os esforços para a difusão e uso de *software* livre vão desde iniciativas isoladas de educadores que montaram laboratórios de informática para seus alunos até iniciativas em esferas governamentais como o caso do México e do próprio governo federal do Brasil com o SERPRO, que conduz estudo sobre o Linux (ALMEIDA, 2000). Instituições governamentais (nas esferas federal, estadual e municipal) acenam para o uso de *software* livre como forma de diminuição de gastos e inclusão digital (GOVERNO FEDERAL, 2006). Novaes e Coelho (2005) também comentam esse esforço do governo federal e ainda sugerem que as instituições de ensino devem fomentar o desenvolvimento de pesquisas com *software* livre. Entretanto deve-se tomar cuidado com o termo *software* livre, como poderá ser observado mais adiante.

Para explicar um pouco mais é necessário, antes de tudo, categorizar um *software* devido às restrições e direitos impostos, de acordo com o seguinte (REIS, 2003):

- **Software Proprietário:** *software* que proíbe redistribuição e alteração do código pelo usuário. A maior parte dos *software* comercialmente distribuídos hoje se enquadra nesta categoria;
- **Freeware:** *software* que permite redistribuição, mas não modificação, e, portanto para o qual geralmente não há código-fonte disponível. Não é *software* livre;
- **Shareware:** *software* que permite redistribuição, mas que restringe o uso de acordo com uma condição específica, normalmente associada a um tempo limite de uso ou recursos bloqueados, após o tempo ou para desbloquear recursos é necessário adquirir uma licença comercial. Normalmente não há código-fonte disponível para *shareware*;
- **Software Livre:** *software* que oferece ao usuário o direito de usar, estudar, modificar e redistribuí-lo, garantindo as quatro liberdades necessárias e detalhadas a seguir, comentadas também em Bizello e Ruschel (2007). Segundo o movimento *Software Livre* e a *Free Software Foundation* (2006) (FSF) este tipo de *software* leva em conta questões sociais, morais e éticas não

concordando com a cobrança e o “aprisionamento” do conhecimento dentro de um *software* proprietário;

- **Open Source:** este tipo de *software* pretende garantir as quatro liberdades necessárias, como no tipo anterior. Entretanto, o movimento *Open Source*, liderado pela *Open Source Initiative* (2011) (OSI) alega que o *software open source* pode trazer vantagens técnicas (em potencial) e financeiras para as empresas, permite a cobrança pelo *software*, chegando a ser conivente com o *software* proprietário, sendo este o ponto de divergência com o anterior;
- **Domínio Público:** *software* sem *copyright*, cujo proprietário rescindiu qualquer direito que possuía sobre o mesmo, ou ainda *software* cujo *copyright* já expirou. Este tipo de *software* pode ser utilizado sem qualquer restrição.

Existe muita confusão em torno do termo “*software* livre”, tanto que a *Free Software Foundation* (2006) (FSF) mantêm uma página na internet somente para esclarecer sobre frases e termos que é melhor evitar para não haver confusão quanto a esse assunto. Ainda segundo esta entidade *software* livre é uma questão de liberdade e, não de preço. A FSF tem ainda um pensamento que exemplifica bem a questão: deve-se pensar em “liberdade de expressão”, não em “cerveja grátis”. Tais termos e frases se referem à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem *software* sob esta definição. Assim, o termo “*software* livre” não significa “não comercial” (ou gratuito), e por isso é possível um *software* livre disponível para uso e distribuição comercial.

Assim como a FSF, a OSI também defende os direitos de usuários de *software* no sentido de garantir o acesso ao código fonte e sua modificação. A diferença entre esses grupos existe na medida em que a FSF, ou defensores do *software* livre é um movimento social que leva em conta questões éticas e morais. A OSI defende uma relação de mercado contando que a melhoria da qualidade técnica (em potencial) possa trazer benefícios estratégicos e financeiros para empresas de qualquer setor (OPEN SOURCE INITIATIVE, 2011). Desta maneira, a diferença entre esses dois movimentos (FSF e OSI) é a argumentação, aparecendo unidos em diversas situações em favor dos mesmos *software* no contexto da nova sociedade da informação e economia, podendo ser citados de forma agregadora por

meio da sigla FLOSS (*Free/Libre and Open Source Software*) como em estudo da *International Institute of Infonomics* (2005). Esta sigla será utilizada daqui em diante, neste texto, para designar *software* com estas referidas características.

Assim, há um ponto de concordância entre a FSF e a OSI, que é a distribuição do código fonte e, deste ponto de vista, são opostos ao *software* proprietário. Estes *software* tem como maior expoente o sistema operacional Linux. Alguns outros FLOSS vêm se tornando tão popular quanto o Linux como é o caso do BrOffice ou o LibreOffice, que procuram substituir o pacote Office da Microsoft com editor de texto, planilha eletrônica, banco de dados e apresentação de slides. Segundo a FSF, a OSI e o Projeto GNU (2006), FLOSS são aqueles que garantem os quatro tipos de liberdades:

- A liberdade de executar o programa, para qualquer propósito (liberdade nº0);
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade nº1);
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº2);
- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº3).

Para que as liberdades sejam atingidas o acesso ao código fonte é pré-requisito para as liberdades de número 1 e 3. Todas essas liberdades têm de ser irrevogáveis para que o *software* seja realmente FLOSS.

Sabe-se também que a utilização de recursos computacionais em escritórios de projetos de Arquitetura, Engenharia e Construção (AEC) é uma realidade. Utilizam-se editores de texto, planilhas eletrônicas e sistemas CAD (*Computer Aided Design*) para confeccionar documentação técnica e legal. Entretanto, a

utilização dos recursos de informática no setor da construção civil tem um uso menor de tais ferramentas se comparado com a indústria da manufatura (NASCIMENTO; SANTOS, 2003). O incremento do uso das ferramentas computacionais no setor cresceu muito devido ao aumento do poder de processamento dos computadores, a diminuição relativa dos preços e o surgimento de aplicativos direcionados ou gerais (ANDRADE, 2004). Como já foi mostrado os *software* utilizados tem diversas categorias e podem cobrar ou não pelo seu uso e distribuição. Na sua grande maioria, os *software* utilizados em AEC são proprietários e em alguns setores, como o de projetos, de valores altos e necessários em várias estações de trabalho.

O setor da construção civil também começa a procurar e utilizar programas FLOSS, a exemplo do BrOffice, para economizar em investimentos de informática. A possível economia gerada pode ser revertida para investimentos em hardware, por exemplo. Entretanto, atualmente os usuários de sistemas CAD no Brasil e no mundo carecem de um programa do tipo FLOSS que possa ser equiparado ao *software* proprietário padrão e líder de mercado, o AutoCAD®, da Autodesk. Existem opções alternativas a venda (de menor custo) contendo ferramentas similares e até mesmo comandos e ícones de comando muito parecidos e em posições padrão semelhantes, procurando desta maneira seduzir o usuário por meio do mínimo tempo gasto com a transição ou migração de um sistema para o outro. Porém, como foi dito são alternativas econômicas e não FLOSS nos conceitos apresentados. As opções em FLOSS do tipo CAD não são em tão grande quantidade quanto à oferta dos *software* proprietários e, numa primeira vista além de graus de evolução diferenciados tem também um desempenho regular (NOVAES; COELHO, 2005).

3. METODOLOGIA

Para desenvolver este estudo alinhado à citada Lei 11 N° 11.888 (BRASIL, 2008) e atento às expectativas de Novaes e Coelho (2005) seria preciso identificar um *software* do tipo FLOSS, a fim de não onerar quaisquer usuários da evolução do AUTOMET (RUSCHEL; RODRIGUES, 2002). Desta maneira esta pesquisa caracteriza-se como descritiva sendo desenvolvida nas seguintes etapas: levantamento, categorização, triagem, definição do sistema CAD de controle, definição de elementos e critérios de avaliação, aplicação da avaliação e análise. A primeira das etapas tratou-se de uma pesquisa para relacionar *software* do tipo FLOSS. Esta etapa foi cumprida com buscas na *World Wide Web* (WWW) por meio de sistemas de busca, e palavras chave. Completando esta etapa, os variados tipos de *software* foram categorizados. Posteriormente a essa etapa, a triagem dos resultados da pesquisa eliminou os sistemas que não se encaixaram no perfil necessário, ou seja, não garantiam os quatro tipos de liberdades para FLOSS (PROJETO GNU, 2006).

Para se encontrar um sistema CAD com características FLOSS que permita a implementação e a evolução da ferramenta de projeto AUTOMET, foi necessário ter um elemento de controle, ou seja, compará-lo a um *software* que tivesse um conjunto mínimo de ferramentas 2D, 3D e programação. Assim, em outra etapa confirmou-se o *software* proprietário AutoCAD® como o mais utilizado em AEC, por meio de estudos de Freitas e Ruschel (2000) e Tse, Wong e Wong (2005) além das constatações de Samuelson (2008), Souza et al (2009) e Attia et al (2009). Buscando garantir os recursos mínimos necessários e já utilizados na versão atual do AUTOMET, a etapa seguinte ocupou-se de especificar as seguintes avaliações: a análise dos recursos para desenho bidimensional (avaliação 2D), a análise dos recursos para modelagem tridimensional (avaliação 3D) e análise do suporte a programação para automação (extensibilidade). Adotou-se critérios de pontuação. A etapa seguinte foi aplicar efetivamente as avaliações (2D, 3D e extensibilidade), que consistiu em comparar comandos ou métodos básicos dentro de um grupo de mesma natureza do sistema CAD proprietário de controle com os mesmos existentes dos *software* FLOSS. Finalmente, a pontuação de cada *software* FLOSS foi levantada permitindo a análise dos mesmos.

3.1 Sistemas CAD FLOSS

Depois das diretrizes definidas, é possível começar a identificação de sistemas CAD do tipo FLOSS disponíveis. Além de procurar diretamente por sistemas deste tipo, já conhecidos e comentados no meio de trabalho e pesquisa, houve também a pesquisa na *World Wide Web*, por meio do sistema de busca Google. Utilizando-se palavras chave que pudessem trazer resultados positivos tais como: *free cad system* e outros termos correlatos. Foram obtidas diversas páginas de resultado. Com o resultado da pesquisa, a próxima fase foi acessar tais páginas de resultado e avaliar inicialmente a descrição na própria página do *software*. Mesmo com a pesquisa visando encontrar sistemas CAD FLOSS para trabalhar no sistema operacional Linux, sistemas para outras plataformas não foram ignorados.

O sistema operacional Linux escolhido foi o Fedora 5. Nessa distribuição de sistema operacional a atualização de *software* é feita facilmente por meio do gerenciador de pacotes que oferece a escolha do aplicativo desejado previamente pesquisado para a instalação. Tal facilidade ajuda muito o usuário iniciante com Linux. Entretanto alguns sistemas CAD livres podem não constar nas pesquisas feitas por este gerenciador de pacotes. Isso pode ocorrer devido ao sistema CAD não ter sido compilado para essa distribuição. Neste caso é possível que a melhor opção seja baixar o código fonte do sistema CAD e executar sua compilação para o sistema instalado no computador. Desta maneira parte da pesquisa ocorreu também com o sistema operacional livre Ubuntu, então na versão 7.10. A compilação do sistema CAD pode ser uma tarefa não muito comum para a maioria dos usuários de sistemas CAD. Levou-se em conta também que muitos usuários de sistemas CAD trabalham sob a plataforma Windows®. Assim, admitindo que o *software* deve ter aparência e recursos, independente do sistema operacional no qual são executados, testes com as versões para Windows® também são válidas.

3.2 Triagem dos Sistemas CAD FLOSS

Após as pesquisas via *Web*, numa primeira análise por meio das descrições nas páginas

dos próprios sistemas relacionados, foi gerada uma listagem filtrada de possíveis sistemas CAD do tipo FLOSS. Do estudo realizado nesta etapa da pesquisa foram levantados dezessete sistemas CAD apresentados no Quadro 1, que possui uma relação de diversos FLOSS do tipo CAD como *software* livre, *freeware*, *shareware*, *open source* e outros alternativos ao AutoCAD®. Alguns chamam a atenção pela

característica mista além de ser *open source* ou *freeware*, possuir também uma versão comercial. *Software* desta natureza também foram relacionados por possuírem a versão FLOSS. A ordem do Quadro 1 é alfabética e não corresponde a qualquer classificação por quaisquer recursos existentes nos mesmos ou critério de utilização.

Quadro 1 – Listagem de Sistemas CAD encontrados na WWW. Fonte: WWW (2011)

CAD	Licença	Sistema operacional	URL's com acessos verificados e atualizados em 25/03/2011
Arquimedes	<i>opensource</i> e <i>freeware</i>	Windows, Linux, Mac OS, Unix	http://sourceforge.net/projects/arquimedes/
Blender-CAD	<i>opensource</i> e <i>freeware</i>	Windows, Linux, Mac OS, Unix	http://projects.blender.org/projects/blendercad/
BRL-CAD	<i>opensource</i> e <i>freeware</i>	Windows e Linux	http://BRL-CAD.org/
CadStd Lite	<i>freeware/comercial</i>	Windows 95 até XP	http://www.cadstd.com/
Design Workshop Lite	<i>freeware</i>	Windows, Mac OS	http://www.artifice.com/free/dw_lite.html
Fandango	<i>free python scripted</i>	Linux	http://sourceforge.net/projects/elcad/
Minos v 2.1	<i>freeware</i>	Windows 95 até XP	http://www.le-boite.com/minos.htm
MScad	<i>Freeware e comercial</i>	Windows	http://www.mscad.com.br
Pyhtoncad	<i>opensource e freeware</i>	Linux	http://www.PythonCAD.org/
oCADis	licença GPL	Linux , Solaris	http://www.nocrew.org/software/ocadis/
Progecad	<i>Freeware e comercial</i>	Windows, Mac OS	http://www.progesoft.com/en/smart-2009
Qcad	<i>opensource / comercial</i>	Windows, Mac OS, Linux e Unix	http://www.ribbonsoft.com/qcad.html
Sagcad	<i>open source e freeware</i>	Linux	http://sourceforge.net/projects/sagcad
SketchUp	<i>shareware e comercial</i>	Windows, Mac OS	http://SketchUp.google.com/
SoftCAD.3D Lite Version 1.16	<i>freeware</i>	Windows	http://softcad-international.software.informer.com/
Thinking 2D	<i>freeware</i>	Windows	http://www.buildwise.org/library/design/design-aids/thk2d301-cad.htm
Varkon	<i>comercial, freeware e opensource</i>	Windows, Unix e Linux	http://www.varkon.org/

Da listagem apresentada no Quadro 1 alguns sistemas CAD não preenchem os requisitos essenciais de um *software* do tipo FLOSS, ou seja, não garantem as liberdades mencionadas pelo Projeto GNU(2006). Os sistemas CAD que atendem estes requisitos são: BlenderCAD, BRL-CAD, PythonCAD, QCad e Varkon. Dentre estes, os *software* QCad e PythonCAD foram escolhidos para serem analisados no que diz

respeito às ferramentas básicas para desenho bidimensional ou 2D. Os demais foram excluídos devido a ser um aplicativo dentro de outro programa (BlenderCAD) ou a complexidade já na fase de instalação, tornando-se uma tarefa árdua para usuários menos experientes em Linux (BRL-CAD e Varkon).

3.3 Sistema CAD proprietário de controle

Para cumprir a tarefa de identificar os sistemas CAD FLOSS que possam ser considerados aptos ao uso do setor de desenvolvimento de projetos para AEC e, também implementar ferramentas para esse setor, é preciso antes de tudo identificar qual programa ou *software* proprietário tem maior uso dentre os usuários do setor. Isso proporciona que seja atingido um número maior de usuários interessados em usufruir das conclusões deste estudo, sem perder o legado. Para isso utilizou-se da proximidade dos autores com as áreas em questão, ou seja, tanto AEC como sistemas CAD em geral, além dos estudos de Freitas e Ruschel (2000) e Tse, Wong e Wong (2005) que apontam uma utilização de 59% e 93%, respectivamente, do produto AutoCAD®, da Autodesk como *software* convencional de CAD. Mais recentemente, Samuelson (2008) também apontou, em sua pesquisa comparativa, este mesmo *software* proprietário, como o mais utilizado entre arquitetos e técnicos consultados. O AutoCAD® também foi mencionado como *software* mais utilizado para modelagem 3D como indica Attia, et al, (2009, p.207). No cenário nacional Souza et al (2009) identificam o grande uso do AutoCAD® em escritórios de arquitetura de pequeno porte de Curitiba. Tais constatações confirmam a escolha deste *software* como sistema CAD de controle para o estudo em questão.

3.4 Definição de elementos comparativos

Chama-se de ferramenta um comando ou um método existente no sistema CAD. Um comando pode ser, por exemplo, o comando *line* que ao ser digitado, acionado por ícone ou menu dispara uma série de questionamentos e ações que culminam na apresentação de um elemento geométrico linha. Já um método é um recurso que possibilita, por exemplo, a organização dos elementos em camadas (*layers*) ou pode associar valores e

representações diferentes às cotas de um desenho no caso dos estilos de dimensionamento. Normalmente para ter acesso à interface de criação e configuração de tais recursos digitamos o nome do recurso, aciona-se um ícone ou o escolhe-se em um menu *pull-down*.

A princípio identificar ferramentas básicas parece uma tarefa fácil, entretanto é sabido que o nível de utilização dos usuários varia muito de acordo com a experiência na área de atuação e no uso do sistema CAD. Vale lembrar que o sistema base escolhido tem ferramentas que se confundem com os conceitos envolvidos com a qualidade dos arquivos CAD (PANIZZA; RUSCHEL, 2005).

De uma maneira complementar, a aparência do *software*, ou seja, sua interface com o usuário também terá uma análise inicial. Todas as ferramentas e pontos analisados serão comparados ao AutoCAD® e tabulados com valores associados aos seguintes critérios:

- Igual; quando o recurso existir, for identificado facilmente, e se comportar igual ao do AutoCAD®, receberá o valor 3;
- Semelhante; quando o recurso existir, porém não se apresentar claramente e ou possuir pequenas diferenças do AutoCAD®; receberá valor 2;
- Diferente; quando o recurso existir, ter dificuldade de ser identificado e ou for totalmente diferente do AutoCAD®, receberá valor 1;
- Sem avaliação; quando o recurso não existir ou for insuficiente para o que se destina, receberá valor 0 (zero).

O fator humano é um ponto crítico nas mudanças de rotina de trabalho, principalmente no setor da construção civil. Assim é válido ressaltar que quando o usuário está familiarizado com um programa e se encontra em um nível de utilização de intermediário à avançado, se torna difícil pensar nas ferramentas básicas e seu uso restrito, sem levar em conta o projeto como um todo e o uso conjunto e sequencial de outras ferramentas.

4. UMA VISÃO DOS SISTEMAS CAD FLOSS

Nesta seção serão apresentadas a descrição dos sistemas CAD FLOSS escolhidos na etapa de triagem deste estudo.

4.1 O software QCAD

O Qcad é um sistema CAD produzido pela Ribbonsoft, uma sociedade de engenharia suíça, e tem diversos colaboradores pelo mundo (RIBBONSOFT, 2007). Funcionando em múltiplas plataformas a versão Linux foi a testada para este trabalho. Fazendo parte do movimento do *software* livre e com acesso ao código fonte, qualquer programador pode inserir novas funcionalidades. Interessante que este é um caso que mesmo sendo um *software* livre, garantindo as liberdades necessárias, existe uma versão comercial disponível para o sistema operacional Windows®.

Iniciando a análise, ao ser executado na versão 2.05.0 para Linux, sua interface simpática apresenta barras de ferramentas, ícones e menus e um *prompt* de comando por meio dos quais executam-se os comandos necessários e informam-se parâmetros (Figura 1). O *prompt* tem foco de acesso pela barra de espaço. A barra de status acomoda a visualização de coordenadas e controle de objetos selecionados. Os menus têm nomes e os ícones imagens sugestivas. A área de trabalho possui *grid* que pode ser ativado e desativado e tem seu espaçamento configurável. Existem comandos para se iniciar um arquivo novo ou abrir um arquivo de intercambio do tipo DXF (*Data Exchange Format*).

Percebe-se que a interface mais elaborada confere ao programa um olhar mais atento

por parte do usuário que se sente mais motivado para conhecer o *software*.

Começando a análise nota-se que há apenas um ambiente de modelagem e impressão, diferente do *Model Space* e do *Paper Space* do AutoCAD®. Em apoio ao ato de desenhar com o Qcad é possível solicitar pontos de atração em diversos pontos dos objetos. As opções de zoom são boas e aparecem na barra de ferramentas, por padrão.

Quanto à organização do desenho suas ferramentas são acima do regular, pois as unidades podem ser configuradas com precisão para ângulos também. As divisões em camadas são possíveis permitindo controle da visualização e configuração de cor e tipo de linha para cada camada, individualmente. Os textos são configurados, mas não é possível armazenar estilos nomeados. Isso também ocorre na configuração dos parâmetros de dimensionamento. Os comandos de criação usados para comparação, no Qcad, se parecem muito com os do AutoCAD®. Criar geometria básica com esse *software* livre não é tarefa complicada. Somente a comando *block* é confuso quanto à sequencia de ações para criar um bloco de desenho. O comando *wblock* que cria blocos para utilização em outros arquivos não foi encontrado. Alterar a geometria já não é tão intuitivo quanto desenhar no Qcad, porém as ferramentas funcionam bem e todos os comandos usados como critérios foram encontrados. O ponto que deixa a desejar é a impressão que não oferece configuração além das costumeiras para impressoras comuns. Não há como estabelecer relações das unidades de desenho com a saída para o mundo real, no papel, ajustando assim a escala do desenho.

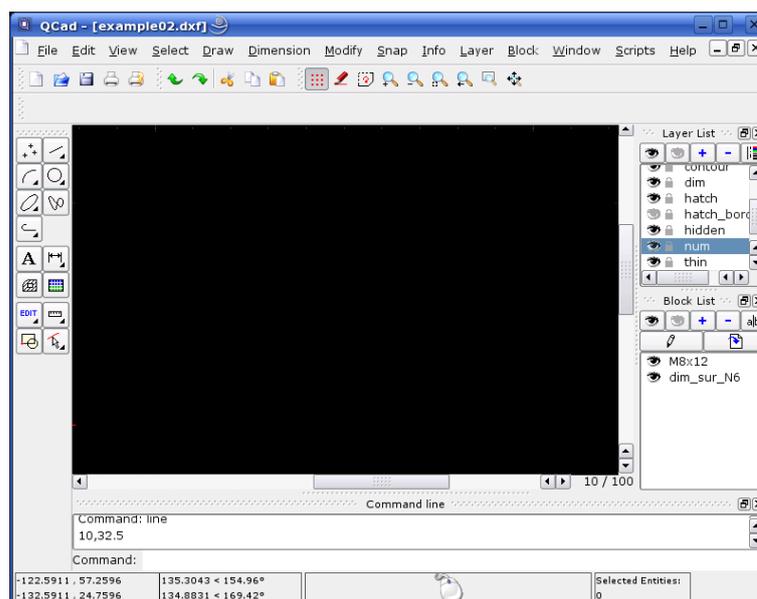


Figura 1 – A tela do Qcad. Fonte: Autores

Como programa que se destina ao desenho 2D, o Qcad oferece ferramentas boas para organizar e executar a geometria que representará os elementos de AEC, entretanto a saída de documentação impressa não acompanha os demais comandos.

4.2 O Software PYTHONCAD

O *software* PythonCAD é um sistema CAD escrito em Python que funciona no sistema operacional Linux, Unix, OSMac, e outros (HAAS, 2007). A linguagem de programação Python garante versatilidade ao sistema. A princípio isso é uma vantagem sendo que programadores podem incluir recursos novos uma vez que tem acesso ao código fonte. Entretanto, programação é uma outra análise, quanto ao desempenho como sistema CAD a análise mostra que este programa tem muito a evoluir ainda. Ao acessar o programa percebe-se sua interface simples (Figura 2). Conta apenas com a barra de menus do tipo *drop down*, um *prompt* de comando e a coluna lateral esquerda para gerenciar as camadas. No meio dessas ferramentas encontra-se a área de trabalho. As coordenadas X e Y são indicadas no lado inferior direito, seus valores mudam conforme a posição do mouse. A falta de barra de ferramentas deixa a aparência do *software* bem diferente do que se espera de um sistema CAD.

No início de uma sessão de trabalho, na caixa de preferências é possível formatar a unidade de trabalho. As ferramentas de organização apresentam todos os itens avaliados com

médio grau de evolução, no geral. É possível organizar o desenho em camadas, mas a aparência dos objetos contidos na camada não é definida ali. Antes de criar o objeto é necessário configurar as propriedades por meio de uma janela de diálogo de preferências. Estilos de textos não podem ser criados e nomeados, apenas suas propriedades são escolhidas antes de efetuar a escrita. Os estilos de dimensionamento também não podem ser nomeados sendo que sua formatação é conservada até que se mude algum parâmetro. O que já foi cotado não muda. Não foi encontrado campo que sirva como multiplicador a ser aplicado ao valor real mensurado.

A geometria pode ser criada com linhas, arcos, círculos e polígonos. Não é possível agrupar elementos e gerar blocos. A ferramenta de texto é interessante e tem boa aparência além de ser considerado texto de múltiplas linhas. Entretanto, não foi encontrada ferramenta que altere o conteúdo do texto, apenas editando propriedades como cor, linha, fonte, etc. Quanto à edição dessa geometria criada é possível dizer que as ferramentas quando existem ou estão disponíveis não são muito intuitivas. Um exemplo é o da ferramenta *mirror* que existe no menu *Edit*, entretanto nunca ficou disponível para uso mesmo com objetos selecionados. Comandos importantes como *Scale* e *copy* não foram encontrados. O comando *move* se aplica a uma entidade por vez ou a todas com *select all*. A entrada de dados utilizando coordenadas totais funciona

assim como no AutoCAD®, porém não foi possível o uso de coordenadas relativas.

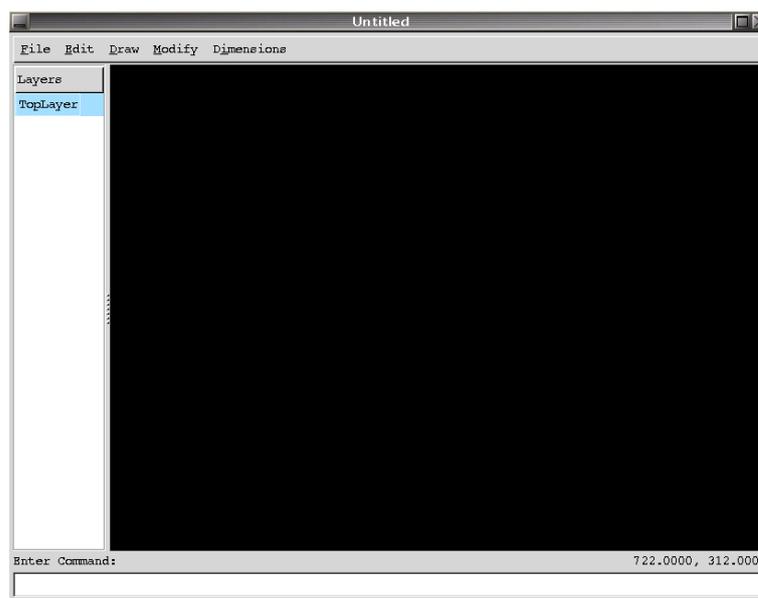


Figura 2 – Tela do PythonCAD. Fonte: Autores

Quanto às ferramentas de apoio, não foi encontrado como configurar um *grid* automaticamente. O *Osnap* é automático e têm como pontos de atração apenas vértices e centros de círculos e arcos. Não foi possível escolher um *midpoint*, por exemplo. Os comandos de *zoom* têm aproximação e redução, possibilidade de escolher uma área de ampliação assim como encaixar todo o limite de desenho na tela. O item das ferramentas de finalização é o mais carente de recursos comparados. Não há ambientes distintos de modelagem e impressão (*model x paper*) e a impressão nos permite fazer um *print screen* e encaixá-lo numa folha padronizada como as de padrão A, por exemplo.

Apesar de nesta versão o PythonCAD se apresentar inferior aos programas CAD proprietários algumas de suas ferramentas são no mínimo interessantes como a ativação de *Split* (variável) que faz com que linhas que se cruzem já sofram uma quebra no ponto de intersecção separando-se em dois objetos. O sistema possui ainda ferramentas existentes no *software* proprietário que não foram comparados como *chamfer*, *fillet* e *break*, este último nomeado como comando *Split* dentro do menu *Modify*.

4.3 O Software BRL-CAD

O BRL-CAD é um aplicativo CAD com modelagem de sólidos em 3D. Optou-se pela versão para o sistema operacional Windows®, visto que se espera que a aparência e os recursos sejam sempre os mesmos, independente do sistema no são executados.

Ao iniciar o programa são apresentadas três janelas. Uma dessas é a janela gráfica onde se pode ver o resultado final. As outras duas são de interação via *prompt*, uma de comandos do programa e a outra uma janela de comando do DOS. Na área gráfica não são encontrados elementos de interação como barra de ferramentas ou ícones, o que faz com que a aparência inicial do programa não seja a esperada pelo usuário familiarizado com o sistema CAD AutoCAD®. Os menus encontrados na janela gráfica e na de comando são os mesmos. É possível também redimensionar e reposicionar as janelas gráficas e de comando, entretanto essas configurações não são gravadas tornando-se necessário fazer os ajustes a cada sessão iniciada. A janela gráfica tem apenas uma divisão e a de comando de *prompt* fica a espera exibindo um cursor na primeira linha, no alto. Não é possível formatar a área de trabalho ou criar objetos sem antes nomear um arquivo. Não foi encontrada uma forma de se montar um *“template”* uma vez que a

formatação do ambiente de trabalho demanda tempo que pode ser considerado perdido a cada novo trabalho iniciado.

Com o arquivo criado é possível então a formatação da área gráfica em janelas

(*multipane*) que conterão cada uma, de início, uma vista diferente do objeto. As vistas de topo, frente, esquerda e perspectiva isométrica fazem parte da apresentação (Figura 3A).

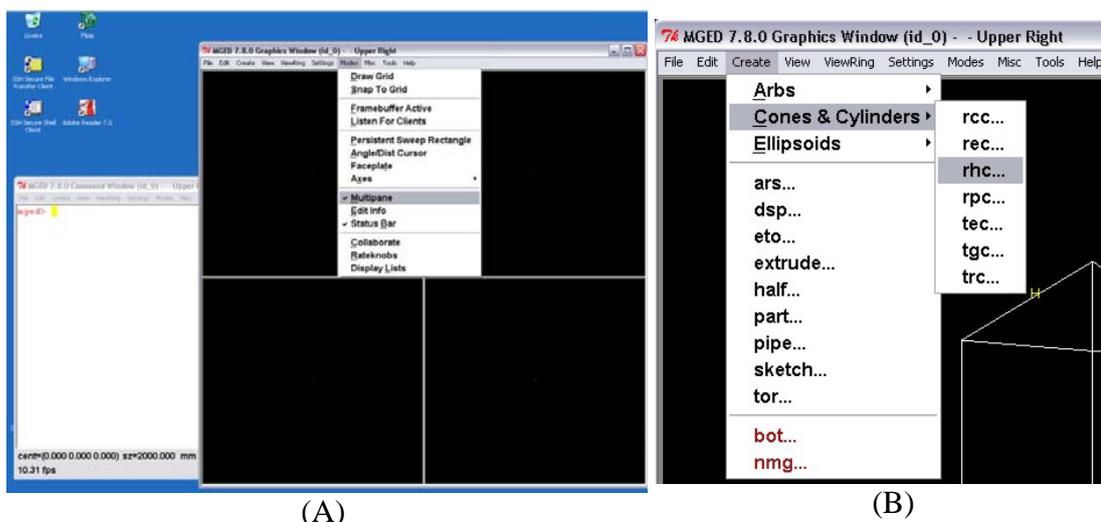


Figura 3 - (A) Tela gráfica do BRL-CAD em *multipane*.; (B) Menu de criação de objetos 3D no BRL-CAD. Fonte: Autores

Após essa divisão do ambiente pode-se escolher qual vista será exibida em que janela. Também pode-se ter a orientação visual por meio de um ícone que indica os eixos x,y e z. Existe um ícone de eixos de visualização e outro de edição. É possível visualizar e configurar o grid, sendo que em algumas configurações o programa se mostrou instável e foi finalizado.

As edições mover, rotacionar e escalar são feitas por meio de sliders que podem ser acionados pelo mouse no menu *Misc/Feceplate* e *Misc/Faceplate GUI*. A criação de objetos é feita por meio do menu *Create* (Figura 3B). Em muitos casos não é possível saber que tipo de objeto será criado.

Ao se explorar o BRL-CAD é possível deduzir que existem diversas ferramentas e o programa tem potencial para a modelagem tridimensional, entretanto sua interface gráfica não seduz o usuário comum de CAD como uma ferramenta intuitiva, de acordo com os novos lançamentos de produtos proprietários existentes ou outros demais de caráter *freeware*. A tarefa de se criar um objeto que se pareça o mais próximo possível de uma habitação com paredes, telhado e

aberturas de portas e janelas além de não intuitiva pode ser frustrante.

4.4 O Software VARKON

O Varkon pode ser encontrado e instalado via Synaptic (instalador automático do Ubuntu) não havendo a necessidade de baixar o código fonte e compilação. Entretanto, após a instalação, o Varkon não aparece nas listas de *software* sendo necessária sua chamada pelo "terminal" (algo como a janela de *prompt* do DOS, no Windows®). Em seguida é solicitado um nome para o projeto e na sequencia um nome para o chamado *job*. Não é possível iniciar um trabalho sem estas duas fases que, também são solicitadas no processo de abrir um arquivo existente que devem estar no diretório padrão criado automaticamente.

No Varkon, assim como o BRL-CAD, encontrou-se dificuldade inicial de se localizar na interface os comandos de configuração e de desenho (Figura 4). O acesso aos comandos é dado por um menu lateral com botões, sendo que sua interface não dispõe de ícones ou menus do tipo *pull-down*.

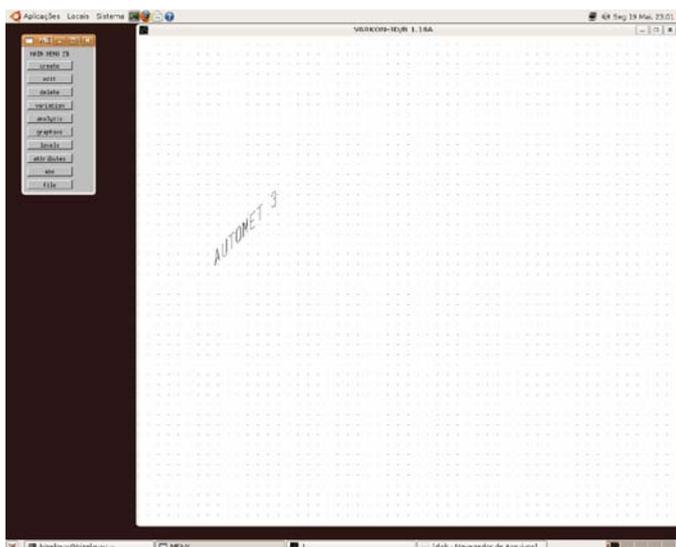


Figura 4 – Tela gráfica do Varkon. Fonte: Autores

Logo após se iniciar um novo projeto pode-se optar por gerar a geometria 2D ou 3D (Figura 5). A partir daí é possível se configurar *grid*, criar *layers* e definir propriedades do objeto

linha. Para se configurar um *layer* é necessário informar um número de nível e depois o seu nome. Entretanto não ficou claro como proceder para desenhar efetivamente.

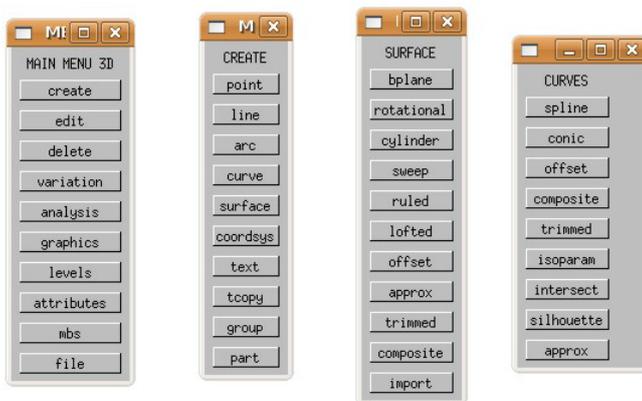


Figura 5 – Menus de modelagem 3D do Varkon. Fonte: Autores

Apesar de mostrar as opções de entrada de dados como coordenadas absolutas ou relativas ou ainda por entrada via cursor, os objetos não foram visualizados. Esperava-se que pela configuração de espessura, tipo e cor de linhas esse problema se resolvesse, entretanto isso não aconteceu. Na opção de desenho via cursor, apesar do *software* solicitar os pontos inicial e final, a linha, que é o resultado esperado, não foi criada após os mesmos serem informados por meio do mouse. O mesmo ocorreu na tentativa de criar algo em 3D. Uma característica do Varkon é a

de trabalhar apenas com superfícies sendo que algumas de suas ferramentas nos remetem a comandos do AutoCAD® tais como *ruled surface*. É possível listar propriedades de objetos 2D e 3D como no AutoCAD®, com o comando *List*.

O Varkon ainda tem recursos que mostram potencial para o uso como CAD para AEC, pois se encontrou sistemas de coordenadas, criação de vistas e calculadora. Entretanto, carece também de apelo visual e comandos mais intuitivos que possam incentivar o usuário a explorá-lo.

5. AS AVALIAÇÕES

5.1 Avaliação 2D

Numa primeira análise pensou-se utilizar a divisão natural do próprio programa existente na barra de menus *pull-down*. Nesses menus encontra-se a possibilidade de formatação de estilos, organização de elementos, criação e modificação da geometria, dimensionamento e impressão além de manipulação de arquivos além da ajuda ou *help*. Encontram-se ferramentas que tem a mesma funcionalidade em menus diferentes. Desta forma foram criados grupos que contém as ferramentas consideradas básicas organizadas por igual funcionalidade. Os comandos ou métodos que dizem respeito à manipulação de arquivos como: *open*, *save*, *close*, etc, não foram incluídos em nenhum dos grupos por serem considerados comuns a qualquer *software*, mesmo que não CAD. Seguem abaixo os itens que nomeiam e explicam cada grupo de ferramentas:

- Ferramentas de Organização – são aquelas responsáveis pela organização, associação e caracterização de elementos independentes de sua natureza. Determinam a unidade de trabalho; podem criar camadas e escolher suas cores, tipos de linhas, visualização impressão e etc; cria estilos de textos definindo tipo de fonte (letra), altura, etc; criar estilos de dimensionamento associando um estilo de texto e definindo características dos elementos da cota, aplicam determinados fatores de conversão e multiplicadores às unidades de desenho.
- Ferramentas de Criação – são responsáveis por criar a geometria que representará os elementos da construção (linhas, arcos, quadriláteros, pontos, círculos, textos, cotas, etc.). Inclui-se também neste grupo os blocos (criação e inserção) que tem a função de agrupar elementos geométricos ou não em um único elemento, desta forma economizam tempo com desenhos repetitivos.
- Ferramentas de Modificação ou Edição – são as ferramentas que modificam o

elemento ou geometria quanto à sua forma parcial ou total e posicionamento. Uma ferramenta que faça uma cópia de um elemento, apesar de criar um novo objeto é considerada uma ferramenta de edição.

- Ferramentas de Apoio – são as ferramentas que auxiliam o trabalho garantindo precisão. Neste grupo é possível incluir os comandos de visualização que aproximam e afastam áreas do desenho para melhor poder trabalhar. Outro recurso deste grupo são os pontos de atração dos objetos que facilitam o dimensionamento, modificação e posicionamento de outros objetos durante a edição.
- Ferramentas de Finalização – são aquelas que estão ligadas ao processo de definição e apresentação do desenho como produto final. Podem ser citados: ambientes de impressão, tamanho de folhas, ajustes de escala, tabela de penas, etc.

Uma vez que as definições de cada grupo foram feitas levando-se em conta a funcionalidade dentro do programa, é preciso especificar quais ferramentas são consideradas básicas dentro de cada grupo. Tal tarefa justifica-se uma vez que o AutoCAD®, a cada versão, foi incrementado com diversos e sofisticados comandos. É possível que os sistemas CAD FLOSS não tenham todos os mesmos comandos, entretanto devem ser capazes de gerar elementos simples com os quais possamos obter geometrias mais complexas, textos e dimensionamentos. Como exemplo tome-se por meio das ferramentas de criação do menu *Draw* do AutoCAD®, no referido menu encontram-se mais de vinte ferramentas que os sistemas CAD FLOSS podem não possuir. Com esse objetivo segue abaixo o Quadro 2 contendo os grupos e as respectivas ferramentas, baseadas no AutoCAD®, que serão avaliadas:

Quadro 2 – Grupos de Ferramentas e comandos básicos do AutoCAD®

Grupo	Comandos ou métodos
Aparência / interface	Ícones, <i>menus</i> , <i>prompt</i> , tela
Ferramentas de Organização	<i>Dimstyle</i> , <i>layer</i> , <i>style</i> , <i>units</i>
Ferramentas de Criação	<i>Arc</i> , <i>block</i> , <i>circle</i> , <i>dimension</i> , <i>line</i> , <i>text</i> e <i>wblock</i>
Ferramentas de Modificação ou Edição	<i>Copy</i> , <i>ddedit</i> , <i>erase</i> , <i>mirror</i> , <i>move</i> , <i>rotate</i> e <i>scale</i>
Ferramentas de apoio	<i>Grid</i> , <i>osnap</i> e comandos de <i>zoom</i>
Ferramentas de Finalização	<i>Plot</i> e <i>tilemode</i>

Na avaliação 2D (Tabela 1) o Qcad aparece ainda como primeira opção mesmo com a

inserção do Varkon e BRL-CAD, pois estes últimos obtiveram avaliações fracas.

Tabela 1 – Avaliação 2D dos CAD FLOSS

		Análise de Programas CAD livre.				Recursos para Desenho 2D																							
		igual (=)	semelhante (≈)	diferente (≠)	sem comparação (∅)																								
		3	2	1	0	FERRAMENTAS														TOTALS									
CAD	area de trabalho	Interface	Organização	Criação						Edição						Apoio	Finalização	PONTUAÇÃO	PORCENTAGEM EM RELAÇÃO AO AUTOCAD										
		ícones	menu	prompt	dymstyle	layer	style	units	arc	block	circle	dimension	line	text	wblock	copy	ddedit			erase	mirror	move	rotate	scale	grid	osnap	zoom	plot	tilemode
Qcad	3	2	2	2	2	2	1	3	3	1	3	2	3	3	0	2	3	2	2	2	1	2	3	2	2	1	0	54	66.67%
Pythoncad	2	0	2	2	1	2	1	2	2	0	3	1	2	2	0	0	0	3	0	2	2	0	0	2	2	1	0	34	41.98%
Varkon	1	0	1	0	1	1	0	1	1	0	0	1	1	1	0	0	1	2	0	0	0	0	1	1	0	1	0	15	18.52%
BRL-CAD	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	7	8.64%

5.2 Avaliação 3D

Na análise de ferramentas de desenho tridimensional (Tabela 2) foram comparados comandos agrupados de acordo com sua natureza, assim como na avaliação 2D. Logicamente foram procurados comandos intimamente ligados ao desenho 3D nos

grupos de criação e modificação (Quadro 3). Muito importante ao desenho 3D, comandos para a visualização do modelo também foram procurados e comparados e, como no 2D, foram inseridos no grupo de Ferramentas de apoio. Deve-se salientar que as ferramentas 2D também são utilizadas em 3D porém não foram avaliadas novamente.

Quadro 3 – Grupos de Ferramentas para 3D

Grupo	Comandos ou métodos
Ferramentas de Criação de	Primitivas geométricas, extrusão e revolução.
Ferramentas de Modificação ou Edição	União, subtração, intersecção e <i>slice</i>
Ferramentas de Apoio	Múltiplas janelas, perspectivas isométricas e ponto de fuga e aparência do modelo (<i>wireframe</i> , <i>shaded</i> , <i>hidden</i> , etc)

Nesta análise, apesar das ferramentas encontradas o BRL-CAD, assim como o Varkon, tem uma desvantagem, sua interface não intuitiva e dificuldade em criar os desenhos volumétricos básicos para AEC. A análise desses sistemas CAD não procura apenas detectar a presença ou não do critério (ferramentas e recursos), mas também verificar sua facilidade de uso em comparação com o líder proprietário de mercado. Isto quer dizer que uma interface intuitiva e com bom visual agregam um apelo que fazem com que o

usuário explore com curiosidade as ferramentas que também devem além de obviamente funcionar corretamente, existirem em número e funcionalidade mínimas.

A Tabela 2 mostra os resultados da avaliação de ferramentas para modelagem 3D e pontua o BRL-CAD como o mais apto dentre os avaliados. O Varkon ficou com a pontuação muito baixa pelo fato de não trabalhar com sólidos assim como o BRL-CAD e o AutoCAD® (elemento comparativo).

Tabela 2 – Avaliação 3D dos CADs FLOSS

Critérios	Análise de Programas CAD Livre				Recursos 3D							
	igual (=)	semelhante (≈)	diferente (≠)	sem comparação (Ø)								
	Valores	3	2	1	0							
CAD	FERRAMENTAS						TOTAIS					
	Desenho Tridimensional											
	Criação		Modificação		Visualização							
		Primitivas	Revolução	Extrusão	União	Subtração	Intersecção	PONTUAÇÃO	PORCENTAGEM EM RELAÇÃO AO AUTOCAD			
BRL-CAD	1	0	1	1	1	0	2	2	2	2	13	39%
Varkon	0	0	0	0	0	0	0	1	0	0	1	3%

5.3 Avaliação Sobre Programação (Extensibilidade)

Para avaliar este aspecto tomou-se como base o recurso de extensibilidade do AutoCAD® que foi utilizada na programação do AUTOMET, o AutoLISP. Desta maneira os principais pontos a serem avaliados são: um ambiente de programação (que inclui testar partes do código pelo *prompt*) e a capacidade de se invocar comandos do programa dentro do código.

Quanto a programação, o PythonCAD foi melhor avaliado, devido ao seu ambiente próprio de programação em Python e a possibilidade de chamar comandos do PythonCAD durante a programação. Porém, como não possui recursos de desenho 3D, será descartado. Entretanto, nos demais *software*, BRL-CAD e Varkon, que possuem elementos 3D, foi possível apenas identificar a linguagem de programação, mas não a possibilidade de executar um respectivo comando numa linha de código, como é desejado.

Tabela 3 – Avaliação do suporte a programação dos CAD FLOSS

Análise de Programas CAD Livre			Programação		
Critérios	igual (=)	semelhante (≈)	diferente (≠)	sem comparação (∅)	
Valores	3	2	1	0	
CAD	FERRAMENTAS			TOTAIS	
	Linguagem de Programação	Linha de Comando	Ambiente de Programação	PONTUAÇÃO	PORCENTAGEM EM RELAÇÃO AO AUTOCAD
PythonCad	Python	2	3	5	83%
Varkon	MBS	0	1	1	17%
BRL-CAD	C	0	0	0	0%
Q-CAD	C++	0	0	0	0%

6. DISCUSSÃO

Como pode ser visto, cada avaliação foi atendida por um sistema CAD FLOSS diferente. A avaliação 2D indica o Qcad como um bom sistema para tal uso pois, mostrou-se mais evoluído e mais completo em recursos e interface simpática pronto para o desenvolvimento de projetos bidimensionais. Em recursos 3D o BRL-CAD aparece como opção razoável em sistemas CAD FLOSS e com grande potencial. Na avaliação de extensibilidade o PythonCAD foi o melhor avaliado apesar de ser mais “limpo” na sua interface. Isso, à princípio, sugere que tem

menos ferramentas do que dispõe, entretanto devido à sua linguagem de programação orientada a objetos, o PythonCAD tem potencial para ser um CAD FLOSS base para a evolução de aplicativos específicos.

De uma forma conjunta (com as três avaliações) os sistemas CAD livres avaliados (QCAD, PythonCAD, VARKON e BRL-CAD) não corresponderam ao esperado. Apenas as qualidades individuais demonstradas, não são suficientes se comparados com o sistema CAD proprietário de referência, nos recursos de 2D, 3D e extensibilidade (Tabela 4).

Tabela 4 – Resumo das Avaliações

Resumo das Avaliações em %			
Avaliação	2D	3D	Extensibilidade
Sistema CAD			
BRL-CAD	8.64%	39%	0%
PythonCad	41.98%	n/f	83%
Qcad	66.67%	n/f	0%
Varkon	18.52%	3%	17%
n/f = não possui ferramenta desta natureza			

É possível dizer que ainda não há um *software* do tipo FLOSS que possa abrigar a ferramenta AUTOMET, pois há a necessidade de desempenho mínimo em 2D, 3D e extensibilidade.

Assim, frente a essa realidade e considerando que este estudo subsidia o futuro desenvolvimento da ferramenta AUTOMET, procurou-se uma alternativa intermediária ao CAD proprietário e FLOSS testados até aqui.

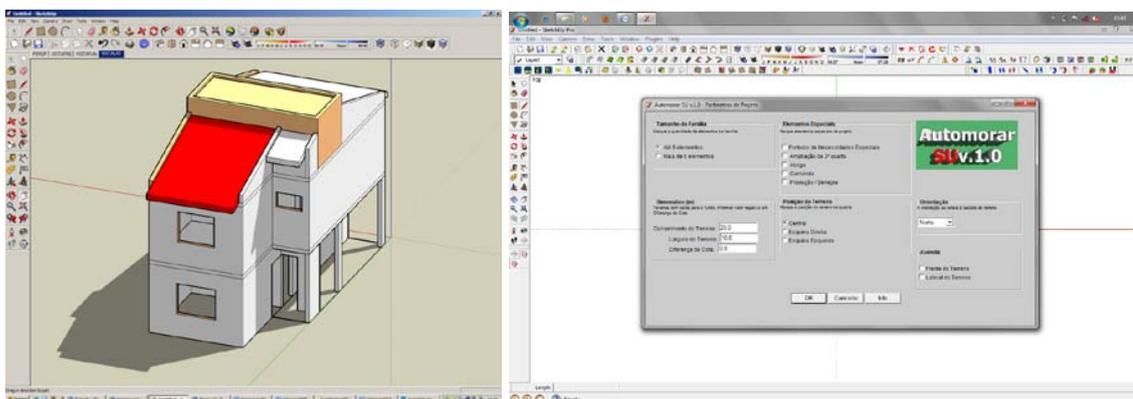
Tal alternativa aparece na popularidade e no crescente uso, por alunos de graduação de arquitetura e mesmo uso profissional (ATTIA et al., 2009; SOUZA et al., 2009; GOOGLE DOCS, 2008) de um programa de características mistas, tendo uma versão *shareware* e uma comercial, o SketchUp. Optou-se então pela versão *shareware* do SketchUp.

7. A OPÇÃO SKETCHUP

O SketchUp não preenche os requisitos de *software* livre segundo Projeto GNU (2006). Este programa tem duas versões: uma primeira que é *shareware* (portanto gratuita) e outra versão profissional (com custo). A versão *shareware* possui menos recursos do que a versão profissional. O SketchUp não possui versão para linux, além disso seu código fonte também não está disponível para que possa ser estudado, alterado e etc. Entretanto, pode ser estendido com pluggins escritos por meio de programação na linguagem Ruby e de sua *Application Programming Interface* (API) do SketchUp. Estas novas funcionalidades sim, acrescidas ao SketchUp, podem ter o seu código fonte disponível para estudo, modificação e distribuição.

O SketchUp (GOOGLE SKETCHUP, 2009) é um programa inovador na forma de desenhar

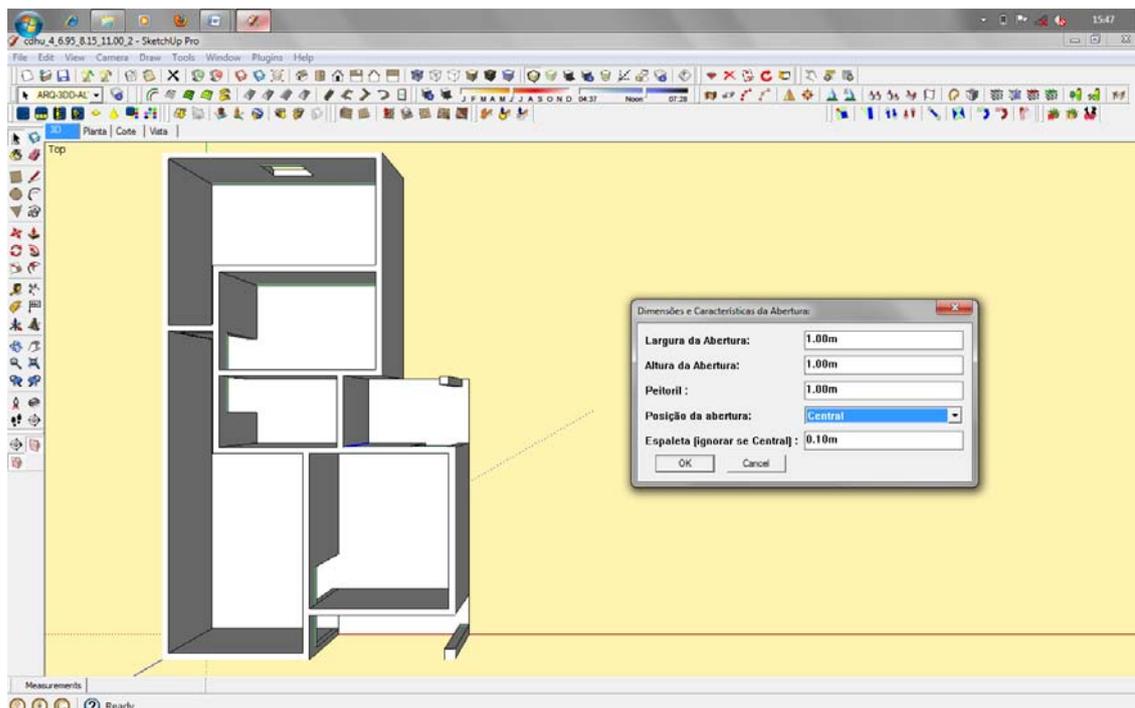
tridimensionalmente. Sua interface é agradável e intuitiva (Figura 6A). A versão *shareware* do programa é extremamente popular entre alunos de arquitetura. Ainda no cenário nacional Souza et al (2009) nos falam sobre o seu uso em escritórios de arquitetura de pequeno porte de Curitiba. Já fora do país, Attia, et al (2009) comentam que o SketchUp juntamente com o AutoCAD®, ultrapassam os demais concorrentes no que diz respeito à utilização para modelagem em 3D. Assim, a utilização acadêmica e os trabalhos profissionais mostram sinais de que suas ferramentas e formas de visualização enriquecem a documentação de projeto, nas suas diversas fases (GOOGLE DOCS, 2008). Para se criar um primeiro modelo, o usuário de CAD, não demora mais do que alguns minutos.



(A)

(B)

Figura 6 – (A) Base de projeto AUTOMET no SketchUp; (B) Teste de interface no SketchUp; (C) Teste de desenvolvimento no SketchUp. Fonte: Autores



(C)

Figura 6 – (A) Base de projeto AUTOMET no SketchUp; (B) Teste de interface no SketchUp; (C) Teste de desenvolvimento no SketchUp. Fonte: Autores (continuação)

O SketchUp possui recursos 2D e 3D intuitivos e de uso relativamente fácil. Possui também capacidade de realizar atualizações na geometria bidimensional conforme o 3D é modificado. É possível importar arquivos nos formatos DWG, DXF, 3DS, JPG, TIFF e etc. No quesito extensibilidade, pode-se programar em Ruby em conjunto com a API do SketchUp. As janelas de diálogo podem se utilizar de HTML e Java Script para interação dos usuários (Figura 6B). Há um console onde é possível experimentar comandos e métodos Ruby antes de inserir o código no arquivo fonte principal. É possível também comunicar-se com banco de dados. Desta maneira tem-se acesso a objetos e métodos que permitem criar e manipular as entidades, o que permite construir aplicações para serem usadas no SketchUp, automatizando rotinas. Algumas de

suas próprias ferramentas, como por exemplo, Sandbox (modelador de terrenos) são feitas nesta linguagem. Assim as tabelas a seguir mostram a pontuação e a boa colocação em comparação com o AutoCAD®, mostrando-o como uma boa alternativa aos objetivos deste trabalho.

Na avaliação 2D (Tabela 5) o SketchUp mostrou-se bem sendo que algumas de suas ferramentas executam as mesmas tarefas mas, é preciso achá-las. Entretanto a interface permite que isso seja feito de maneira rápida. Aliás, a interface é bem simpática e atraente seduzindo o usuário a explorá-la. Nesta versão shareware não é possível dar saída no desenho em escala pois os recursos de impressão permitem isso somente na versão comercial.

Tabela 5 – Avaliação 2D do SketchUp

		Análise de Programas CAD livre.				Recursos para Desenho 2D																							
		igual (=)	semelhante (≈)	diferente (≠)	sem comparação (∅)																								
		3	2	1	0	FERRAMENTAS														TOTALS									
CAD	Interface	Organização				Criação				Edição				Apoio		Finalização		PONTUAÇÃO	PORCENTAGEM EM RELAÇÃO AO AUTOCAD										
	area de trabalho	icons	menu	prompt	dymstyle	layer	style	units	arc	block	circle	dimension	line	text	wblock	copy	ddedit			erase	mirror	move	rotate	scale	grid	osnap	zoom	plot	filemode
Sketchup	3	3	2	0	1	2	1	2	3	2	3	2	3	2	2	2	3	3	1	2	2	2	0	2	3	1	0	52	64.20%

A Tabela 6 apresenta a avaliação 3D do SketchUp. Neste quesito o programa foi bem melhor avaliado e se mostrou apto à esse tipo de modelagem. Por isso não é de estranhar que seja popular entre alunos de arquitetura e

venha crescendo também no meio comercial como mostram pesquisas nacionais e internacionais (SOUZA et al., 2009; ATTIA et al., 2009).

Tabela 6 – Avaliação 3D do SketchUp

Análise de Programas CAD Livre										Recursos 3D					
Critérios	igual (=)			semelhante (≈)			diferente (≠)			sem comparação (Ø)					
Valores	3			2			1			0					
CAD	FERRAMENTAS										TOTAIS				
	Desenho Tridimensional														
	Criação			Modificação			Visualização								
	Primitivas	Revolução	Extrusão	União	Subtração	Intersecção	Slice	Múltiplas Janelas	Wireframe/Hidden	Isométricas	Ponto de Fuga	PONTUAÇÃO	% EM RELAÇÃO AO AUTOCAD		
	2	2	3	2	1	2	1	2	3	3	2				
	Sketchup													23	70%

A avaliação dos recursos de programação ou extensibilidade do SketchUp mostradas na Tabela 7 revelam que este *software* tem grande potencial para se desenvolver e abrigar rotinas automatizadas. Tais que

podem facilitar a vida dos projetistas. Assim como em alguns outros programas estas extensões e rotinas programadas ganham o nome de *plugin*.

Tabela 7 – Avaliação de extensibilidade do SketchUp

Análise de Programas CAD Livre										Recursos 3D					
Critérios	igual (=)	semelhante (≈)	diferente (≠)							sem comparação (Ø)					
Valores	3	2	1							0					
CAD	FERRAMENTAS										TOTAIS				
	Desenho Tridimensional														
	Criação			Modificação			Visualização								
	Primitivas	Revolução	Extrusão	União	Subtração	Intersecção	Slice	Múltiplas Janelas	Wireframe/Hidden	Isométricas	Ponto de Fuga	PONTUAÇÃO	% EM RELAÇÃO AO AUTOCAD		
	2	2	3	2	1	2	1	2	3	3	2				
	Sketchup	2	2	3	2	1	2	1	2	3	3			23	70%

Esses *plugins* podem ser compilados e desta maneira seu código não fica disponível para estudo, alteração e conseqüente evolução. Entretanto se o mesmo não for compilado e for respeitada as liberdades garantidas aos FLOSS, então se tem um *plugin* que pode ser distribuído, estudado e modificado. A Tabela 8

mostra o resultado geral das avaliações do SketchUp identificando-o como um *software* apto ao desenvolvimento e implementação de ferramentas de projeto, pois apresenta todas as ferramentas necessárias e já utilizadas na atual versão do AUTOMET.

Tabela 8 – Avaliação geral do SketchUp

Resumo das Avaliações em %			
Avaliação	2D	3D	Extensibilidade
Sketchup	64.20%	70%	67%
n/f = não possui ferramenta desta natureza			

8. CONCLUSÃO

Os sistemas CAD do tipo FLOSS à disposição para uso são em número considerável, superando assim a barreira à complexidade inerente deste tipo de programa. Do levantamento desses sistemas CAD, somente alguns atendem às qualificações do Projeto GNU (2006). Os *software* analisados neste trabalho se mostraram inferiores ao líder do mercado CAD proprietário.

Verificou-se que cada sistema CAD tem uma característica mais forte que o habilita para determinado uso, dentro dos aspectos avaliados (2D, 3D e extensibilidade) mas, nenhum deles corresponde ao esperado quando se analisa tais aspectos em conjunto. Entretanto esta introdução ao mundo dos CAD FLOSS deixa boas perspectivas para o setor. Primeiro pelo número de *software* que foi encontrado. Em segundo lugar os perfis dos programas são diferenciados e atendem a todos os gostos.

Independente disso para a análise de um sistema CAD FLOSS sugere-se:

- Avaliar sua interface que contendo menus, barras de ferramentas e mostrando informações facilita o trabalho intuitivamente; uma interface simpática e agradável despertará a curiosidade do usuário para explorar o *software*;
- Verificar as ferramentas de organização que auxiliam gerenciando camadas, estilos de texto e dimensionamento e definindo unidades;
- Avaliar a capacidade do programa em criar geometria básica, textos e dimensionamento (ferramentas de criação). O agrupamento de geometrias em grupos (blocos) é desejável, pois gera produção;
- Verificar as ferramentas de modificação ou edição pois aliadas às de criação em sequencia correta também geram produção;
- Verificar ferramentas de apoio que facilitam a execução do trabalho proporcionando precisão, fácil visualização e navegação pelo documento;

- Verificar ferramentas de construção, modificação e visualização do modelo tridimensional;
- Que o programa permita saídas de impressão com ajuste de escala, de preferência sem precisar re-escalar os elementos do desenho;
- O programa deve permitir automatizar tarefas rotineiras e repetitivas por meio de *scripting* ou linguagem de programação.

O estudo realizado pretende eleger um CAD FLOSS que possa abrigar a implementação da ferramenta de projeto AUTOMET, entretanto isso não foi possível, pois os mesmos, como já explicitados, não satisfazem as três avaliações ao mesmo tempo.

Como opção intermediária aparece o SketchUp, *software* CAD do tipo *shareware*, que se torna popular satisfazendo, inicialmente, a comunidade universitária, que percebeu a facilidade de aprendizado e uso e no setor produtivo que admite seu uso também profissionalmente nas rotinas dos

escritórios de arquitetura nacionais e estrangeiros.

O SketchUp, além de bem pontuado em nossas avaliações, é objeto de demais estudos em desenvolvimento (TREDINNICK, et al., 2006) por sua capacidade de extensão. Assim, se mostra viável sua utilização na pesquisa citada neste trabalho, mesmo na versão *shareware*. Em conjunto com a linguagem de programação Ruby mais SketchUp-API pode ser utilizado na implementação de rotinas de produtividade, metodologias de projeto e etc. A disponibilidade do código de tais rotinas para estudo, aperfeiçoamento e distribuição atende às liberdades exigidas para FLOSS (PROJETO GNU, 2006).

Em todas as áreas, a tecnologia evolui para sistemas que possam integrar as fases de execução de um produto ou empreendimento. BIM (*Building Information Modeling*) aparece neste contexto na indústria da construção. Desta maneira, como já existem exemplos e esforços em CAD livre, o mesmo se espera com BIM *open source* (HARRISON, 2009).

REFERÊNCIAS

- ALMEIDA, R.Q. *Por que usar software livre?*. Disponível em: <<http://www.dicas-l.com.br/dicas-l/20000116.php>> Acesso em: 28/02/2008.
- ANDRADE, C.M.G. Software Livre: Alguns Aplicativos Científicos para Engenharia. In: COBENGE - CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA, 2004, Brasília - DF., 2004.
- ATTIA, S.; BELTRÁN, L.; DE HERDE, A.; HENSEN, J. "Architect Friendly": A Comparison of Ten Different Building Performance Simulation Tools. In: INTERNATIONAL IBPSA CONFERENCE - BUILDING SIMULATION 2009, 11., 2009, Glasgow. Scotland, 2009. p. 204-211. http://www.ibpsa.org/m_bs2009.asp. Acesso em: 23/03/2011.
- BIZELLO, S.A.; RUSCHEL, R.C. CAD LIVRE PARA ARQUITETURA, ENGENHARIA E CONSTRUÇÃO. In: ENCONTRO DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO CIVIL, III, 2007, Porto Alegre. *Anais...* Porto Alegre, 2007.
- BRASIL. *Lei Nº 11.888 de 24 de Dezembro de 2008*. Assegura às famílias de baixa renda assistência técnica pública e gratuita para o projeto e a construção de habitação de interesse social. Brasília
- FREE SOFTWARE FOUNDATION. *Palavras e frases confusas e carregadas que devem ser evitadas*. Disponível em: <<http://www.gnu.org/philosophy/words-to-avoid.pt.html>> Acesso em: 13/05/2006.
- FREITAS, MÁRCIA REGINA DE; RUSCHEL, R.C. *O processo de Projeto de Edificações e Ferramentas Gráficas Utilizadas*. In: SIMPÓSIO NACIONAL DE GEOMETRIA DESCRITIVA E DESENHO TÉCNICO, 14., 2000, Ouro Preto. *Anais...* Ouro Preto: UEMG. v. CD-ROM, 2000. p. 1-7.
- GOOGLE DOCS. *Architecture Case Studies_1*. Disponível em: <http://docs.google.com/View?docid=dc837t9h_22r53r9> Acesso em: 30/10/2008.
- GOOGLE SKETCHUP. *Google SketchUp - 3D for Everyone*. Disponível em: <<http://SketchUp.google.com/product/gsup.html>> Acesso em: 15/10/2009.

GOVERNO FEDERAL. *Portal Software Livre*. Disponível em: <<http://www.softwarelivre.gov.br>> Acesso em: 11/09/2006.

HAAS, A. *Site do software PythonCAD*. Disponível em: <<http://www.PythonCAD.org/>> Acesso em: Março de 2007.

HARRISON, D. *Behind the Building Information Model Buzz*. Disponível em: <http://www.stress-free.co.nz/behind_the_building_information_model_buzz> Acesso em: 08/01/2009.

INTERNATIONAL INSTITUTE OF INFONOMICS. *Free/Libre and open source Software: Survey and Study*. Disponível em: <<http://www.flossproject.org/index.htm>> Acesso em: 25/03/2011.

NASCIMENTO, L.A.; SANTOS, E.T. A Indústria da Construção na Era da Informação. *Ambiente Construído*, Porto Alegre, v. 3, n. 1, p. 69-81, jan./mar. 2003.

NOVAES, C.C.; COELHO, S.B.S. *O Uso de Software Livre na Construção Civil*. In: SIMPÓSIO BRASILEIRO DE GESTÃO E ECONOMIA DA CONSTRUÇÃO, 4., 2005, Porto Alegre. Porto Alegre, 2005.

OPEN SOURCE INITIATIVE. *About The Open Source Initiative*. Disponível em: <<http://www.opensource.org/about>> Acesso em: 22/03/2011.

PANIZZA, A.C.; RUSCHEL, R.C. *Avaliação da Competência Para Colaboração em CAD Associada à Utilização da Tecnologia da Informação no Desenvolvimento do Projeto*. In: SEMINÁRIO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO CIVIL, II., 2005, São Paulo. , 2005.

PROJETO GNU; FUNDAÇÃO PARA O SOFTWARE LIVRE. *O que é software livre?*. Disponível em: <<http://www.gnu.org/philosophy/free-sw.pt.html>> Acesso em: 11/09/2006.

SAMUELSON, O. The IT - barometer - a decade's development of IT use in the Swedish construction sector. *ITCon*, v. 13, p. 1-19, 2008 2008. <http://www.itcon.org/2008/1>. Acesso em: 22/03/2011.

REIS, C.R. *Caracterização de um Processo de Software para Projetos de Software Livre*. São Carlos, 2003. 247 f. Dissertação de Mestrado em Ciências da Computação e Matemática Computacional - Instituto de Ciências Matemáticas e de Computação, USP.

RIBBONSOFT GMBH. *Site do QCad*. Disponível em: <<http://pt-br.ribbonsoft.com/>> Acesso em: Março de 2007.

RUSCHEL, R.C.; FREITAS, M.R. CAD na cabeça. *Revista AU - Arquitetura e Urbanismo*, São Paulo, n. 90, p. 94-95, junho/julho 2000.

RUSCHEL, R.C.; RODRIGUES, A.B.F. *Um Exemplo de Cooperação Para a Aplicação e Adaptação de Ferramenta de Projeto Arquitetônico em Empreendimentos de Habitação de Interesse Social*. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, IX., 2002, Foz do Iguaçu - Paraná. , 2002.

SOUZA, C.R; HILGENBERG, F.B.; SCHEER, S. *A TI no Processo de Projeto em Escritórios de Arquitetura de Pequeno Porte*. In: ENCONTRO DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO CIVIL, IV., 2009, Rio de Janeiro. *Anais...* Rio de Janeiro, 2009.

TREDINNICK, R.; ANDERSON, L.; RIES, B.; INTERRANTE, V. *A Tablet Based Immersive Architectural Design Tool* In: ACADIA CONFERENCE, 2006, University of Kentucky - Louisville, Kentucky. , 2006.

TSE, T.K.; WONG, K. A.; WONG, K.F. The utilization of building information models in nD modelling: A study of data interfacing and adoption barriers. *ITcon* ,Vol. 10, Special Issue From 3D to nD modelling, pg. 85-110, <http://www.itcon.org/2005>.

DADOS DOS AUTORES:

(i) Tecnólogo em edifícios da Coordenadoria de Projetos da Faculdade de Engenharia Civil, Arquitetura e Urbanismo da Universidade Estadual de Campinas (CPROJ-FEC-UNICAMP)| bizello@fec.unicamp.br |CV Lattes: <http://lattes.cnpq.br/5566309011001560>

(ii) Docente e pesquisadora na Faculdade de Engenharia Civil, Arquitetura e Urbanismo da UNICAMP|regina@fec.unicamp.br|CV Lattes: <http://lattes.cnpq.br/5458829100371043>