

**UNIVERSIDADE FEDERAL DA BAHIA  
FACULDADE DE ARQUITETURA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ARQUITETURA E URBANISMO**

ANTÔNIO CARLOS DE SOUSA BRITO

**O SOFTWARE LIVRE GRÁFICO NA ARQUITETURA:  
Uma experiência de avaliação**

Salvador  
julho, 2011

ANTÔNIO CARLOS DE SOUSA BRITO

**O SOFTWARE LIVRE GRÁFICO NA ARQUITETURA:  
Uma experiência de avaliação**

Dissertação apresentada ao PPGAU – Programa de Pós-Graduação em Arquitetura e Urbanismo da Universidade Federal da Bahia, como requisito à obtenção de título de Mestre em Arquitetura e Urbanismo.

Orientador: Antônio Pedro Alves de Carvalho

Salvador  
julho, 2011

## RESUMO

As transformações econômicas e sociais desencadeadas pelo processo de globalização dos mercados, pressionam empresas e profissionais a se inserirem no contexto da automação, como prerrogativa para manterem-se atuantes, face à competitividade gerada com a crescente corrida pela capacitação tecnológica, em resposta às necessidades da sociedade contemporânea. Como reflexo disso, a indústria da informática cresceu exponencialmente e, atualmente, se configura como uma das mais poderosas organizações do sistema de acumulação capitalista, detentora de monopólios baseados na manipulação do conhecimento e na produção da tecnologia. Em contraponto, observa-se que a prática da “pirataria” dissemina-se nos mais diversos meios, relacionada, dentre outros fatores, com a necessidade de modernização dos processos produtivos e o alto custo para obtenção desses recursos. Dentro desse contexto, surge a filosofia do *software* livre, que tem chamado atenção no âmbito do debate mundial acerca do tema da inclusão digital, suscitando a busca de alternativas para o desenvolvimento e a produção de tecnologias acessíveis a todas as camadas sociais. Trazendo essa problemática para o campo da arquitetura, observa-se que a automação do projeto estabeleceu-se de forma sólida tanto no ensino como na prática profissional, visando a inserção do seu contingente produtivo na realidade do mercado. Contudo, as ferramentas computadorizadas de auxílio ao projeto mais difundidas e empregadas são provenientes de empresas desenvolvedoras de *software* com finalidades exclusivamente comerciais, que cobram preços altos, e muitas vezes restritivos, pela licença de utilização desses programas, expondo os usuários em situação irregular a graves sanções legais, atingindo principalmente pequenas corporações, os setores públicos e acadêmicos, bem como usuários autônomos. A característica independente e não uniforme da produção e da divulgação do *software* livre, ao mesmo tempo em que ele se estabelece como alternativa promissora de acesso à informatização, levou a indagar sobre a possibilidade de aplicação de ferramentas dessa natureza no campo do projeto de arquitetura, face às restrições impostas pela indústria do *software* proprietário, mais especificamente das companhias que produzem ferramentas da computação gráfica, amplamente empregadas na prática projetual de arquitetura. Nesse contexto, o estudo iniciou-se com uma abordagem conceitual sobre *software* livre que se estendeu a aspectos de seu desenvolvimento, abrangendo a forma de organização e o perfil das comunidades de programadores e usuários. Seguiu-se a essa abordagem uma análise histórica do desenvolvimento das principais aplicações da computação voltadas para o campo do projeto, desde sua origem, até os dias atuais, como forma de estabelecer um panorama das tecnologias mais difundidas e o estado da arte da automação do projeto. Em seguida identificou-se ferramentas livres enquadradas nessa categoria, com potencial de aplicação na produção de projetos de arquitetura e, utilizando-se de técnicas prescritas pela ABNT – Associação Brasileira de Norma Técnicas, selecionou-se entre elas um conjunto de três programas gráficos: Caduntun, Inkscape e Blender, que foram submetidos a um processo de avaliação qualitativa nos ambientes acadêmico, institucional e corporativo de produção de projetos arquitetônicos, obtendo-se resultados que permitiram formular conclusões e tecer recomendações para o uso e desenvolvimento dessas ferramentas na prática e no ensino da arquitetura.

**Palavras chaves:** *Software* livre, *Software* Gráfico, CAD e Avaliação de *Software*.

## ABSTRACT

The social and economic transformations that have been unleashed by the process of markets becoming globalized, pressure companies and professionals to insert themselves in an automated context as a prerogative to remaining operative in the face of the competitiveness derived from the budding race for technological capacity which exists in contemporary society. As a consequence, the information science industry has grown exponentially and presently, it is one of the most powerful organizations in the capitalist system; monopoly-holder in the manipulation of information and technological production. In contrast, the practice of piracy has spread to a variety of milieus, all related, among other factors, by the necessity of the modernization of the productive process and the high costs of obtaining these resources. It is within this context that the philosophy of Free Software has established itself and in which it has garnered notice in the worldwide debate over digital inclusion as well as sounded a call for the search for alternatives in the development and production of technology that is accessible to all societal levels. In bringing this debate to the field of architecture, we can see that the automation of project design has firmly established itself in both the teaching and professional practice of said field and thereby has driven the insertion of architects into contemporary market reality. However, the most widely-known and used computer-aided design tools originate from software developers with exclusively commercial intentions, and who charge restrictive and expensive prices for the licensing of said programs which in turn place illegal users, usually small-business owners or smaller corporations, the public and academic sector as well as individual and autonomous users, at risk of serious legal sanctions. The independent and changing nature of the production and dissemination of Free Software, while establishing itself as a promising alternative for access to information science technology, also led to the inquiry of its applications to computer-aided design in architecture in spite of the restrictions imposed by the software industry owners, and more specifically of the companies which produce graphic design tools widely-used in architectural project design. And, following the norms prescribed by the ABNT – *Associação Brasileira de Normas Técnicas*, a grouping of three graphic design software programs was chosen: Caduntun, Inkscape and Blender. These three programs were subjected to a process of qualitative evaluation in academic, institutional and corporate environments of architectural project production, obtaining results which allowed for the formulation of conclusions and inclusion of recommendations, for the use and development of these tools in the practice and teaching of architecture.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> - Diagrama de um projeto de <i>software</i> livre em desenvolvimento.....	23
<b>Figura 2</b> - Sketchpad em uso.....	35
<b>Figura 3</b> - IBM PC, versão original.....	39
<b>Figura 4</b> - Lisa.....	40
<b>Figura 5</b> - Apple Macintosh.....	41
<b>Figura 6</b> – Qualidade no ciclo de vida do software.....	66
<b>Figura 7</b> - Modelo de qualidade para qualidade externa e interna.....	67
<b>Figura 8</b> - Níveis de pontuação para a métrica.....	70
<b>Figura 9</b> - Relacionamento entre as normas do processo de e as de suporte para a avaliação.....	71
<b>Figura 10</b> - Processo de avaliação, segundo a NBR ISO/IEC 14598.....	71
<b>Figura 11</b> - Interface gráfica do Caduntun.....	90
<b>Figura 12</b> - Interface gráfica do Inkscape.....	92
<b>Figura 13</b> - Interface gráfica do Blender.....	94
<b>Figura 14</b> - Imagem de modelo construído com o Blender.....	95
<b>Figura 15</b> - Estrutura do GQM.....	104
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	
15.	

## LISTA DE QUADROS

<b>Quadro 1</b> - <i>Steakholders</i> e seus papéis.....	19
<b>Quadro 2</b> - Motivações no desenvolvimento do <i>software</i> livre.....	20
<b>Quadro 3</b> - Composição do sistema CAD.....	44
<b>Quadro 4</b> - Série de normas da ABNT para avaliação de produto de <i>software</i> .....	65
<b>Quadro 5</b> - Definição dos atributos de qualidade em uso.....	68
<b>Quadro 6</b> - Síntese do processo de avaliação do "pacote" de programas gráficos livres... 74	
<b>Quadro 7</b> - Modelo de qualidade para seleção dos programas utilizados no experimento	78
<b>Quadro 8</b> - Propriedades relacionadas com requisitos de Adequação.....	80
<b>Quadro 9</b> - Resultado da pesquisa de <i>softwares</i> livres gráficos na página <i>web</i> sourceforge.net.....	81
<b>Quadro 10</b> - Pré-Seleção <i>software</i> CAD 2D.....	83
<b>Quadro 11</b> - Pré-seleção <i>software Vector Graphic Editor</i> .....	84
<b>Quadro 12</b> - Pré-Seleção <i>software</i> CAD 3D.....	84
<b>Quadro 13</b> - Programas CAD 2D selecionados.....	86
<b>Quadro 14</b> - Programas <i>Vector Graphich Editor</i> selecionados.....	87
<b>Quadro 15</b> - Programas CAD 3D selecionados.....	88
<b>Quadro 16</b> - Modelo de qualidade para avaliação dos programas livres gráficos.....	100
<b>Quadro 17</b> - Métricas propostas.....	105
<b>Quadro 18</b> - Métricas selecionadas.....	106
<b>Quadro 19</b> - Níveis de pontuação para fins de julgamento da avaliação.....	107
<b>Quadro 20</b> -Tempo gasto, por participante, para execução do exercício prático.....	113
<b>Quadro 21</b> - Impressões dos participantes sobre o programa Caduntun.....	115
<b>Quadro 22</b> - Graus de dificuldade atribuídos pelos participantes ao programa Caduntun.....	115
<b>Quadro 23</b> - Impressões dos participantes sobre o programa Inkscape.....	116
16. <b>Quadro 24</b> - Graus de dificuldade atribuídos pelos participantes ao programa Inkscape. 116	
17. <b>Quadro 25</b> - Impressões dos participantes sobre o programa Blender.....	116
18. <b>Quadro 26</b> - Graus de dificuldade atribuídos pelos participantes ao programa Blender...	117

<b>Quadro 27</b> - Tabulação do resultado da medição da características de qualidade do Caduntun.....	120
<b>Quadro 28</b> - Tabulação do resultado da medição das características de qualidade do Inkscape.....	122
<b>Quadro 29</b> - Tabulação do resultado da medição das características de qualidade do Blender.....	124

## LISTA DE ABREVIATURAS E SIGLAS

- (AEC) Arquitetura, Engenharia e Construção
- (CAD) Computer Aided Design
- (CAAD) Computer Aided Architectural Design
- (CAE) Computer Aided Engineering
- (CCSL) Centro de Competência em Software Livre
- (CISL) Comitê de Implementação do Software Livre
- (CONISLI) Congresso Internacional de Software Livre
- (CPM) Critical Path Method
- (CSG) Construct Solid geometry
- (CT-INFO) Fundo Setorial de Tecnologia da Informação do Ministério da Ciência e Tecnologia
- (ENSL) Encontro Nordeste de Software Livre
- (ENSOL) Encontro de Software Livre da Paraíba
- (FAQ) Frequent Asked Questions
- (FASOL) Fórum Amazônico de Software Livre
- (FINEP) Financiadora de Estudos e Projetos
- (FISLI) Festival Internacional do Software Livre
- (FSF) Free Software Foundation
- (FSLDC) Fórum de Software Livre de Duque de Caxias
- (GPL) General Public License
- (IA) Inteligência Artificial
- (ITI) Instituto de Tecnologia da Informação
- (ITS) Incompatible Timesharing System
- (MCT) Ministério da Ciência e Tecnologia
- (MIT) Massachusetts Institute of Technology
- (PC) Personal Computer
- (PERT) Program Evaluation and Review Techniques
- (PSL) Projeto Software Livre
- (PSSL) Programa Serpro de Software Livre
- (SERPRO) Serviço de Processamento de Dados do Governo Federal



## SUMÁRIO

**INTRODUÇÃO****1- SOFTWARE LIVRE: uma alternativa de inclusão tecnológica**

1.1 Breve Histórico.....	11
1.2 Aspectos Conceituais e Filosóficos do <i>Software</i> Livre.....	15
1.3 O Desenvolvimento e o Acesso ao <i>Software</i> Livre.....	17
1.4 Panorama Geral do <i>Software</i> Livre no Brasil.....	25

**2 - EVOLUÇÃO DA COMPUTAÇÃO E PROGRAMAS GRÁFICOS**

2.1 Uso do Computador na Área Gráfica.....	33
2.2 A Popularização do Uso do Computador como Ferramenta de Auxílio ao Projeto.....	38
2.3 Classificação das Ferramentas CAD.....	43
2.4 Ferramentas CAD Genéricas.....	45
2.5 Ferramentas CAD Dedicadas.....	45
2.6 Considerações Sobre a Sistemática do Projeto de Arquitetura.....	47
2.7 Principais Aplicações Para a Área do Projeto de Arquitetura.....	50
2.8 Tendências.....	58

**3 - AVALIAÇÃO DE SOFTWARE: programas livres gráficos na arquitetura**

3.1 Normas de Avaliação de <i>Software</i> .....	63
3.2 O Modelo de Qualidade da NBR ISO/IEC 9126.....	66
3.3 Métricas de Avaliação de <i>Software</i> .....	69
3.4 O Método de Avaliação da NBR ISO/IEC 14598.....	70
3.5 Elaboração da Metodologia de Seleção e Avaliação dos Programas Livres Gráficos...	72
3.6 Especificação dos Requisitos Para Escolha das Ferramentas.....	77
3.7 Execução da Seleção dos Programas Livres Gráficos Avaliados.....	81
3.8 Programas Livres Gráficos Selecionados.....	89

## **4 - UMA EXPERIÊNCIA DE AVALIAÇÃO EM USO**

4.1 Especificação dos Requisitos de Avaliação.....	98
4.2 Ambientes de Aplicação do Experimento.....	101
4.3 Tempo de Duração do Experimento.....	101
4.4 Definição das Métricas.....	103
4.5 Plano de Avaliação.....	108
4.6 Seleção dos Participantes.....	109
4.7 Equipamentos Necessários à Avaliação e Fonte de Acesso às Ferramentas.....	109
4.8 Metodologia de Condução da Avaliação.....	110
4.9 Execução da Avaliação.....	112
4.10 Tempo da Avaliação.....	113
4.11 Resultado das Tarefas.....	114
4.12 Resultados de Satisfação.....	114
4.13 Dificuldades Encontradas.....	117
4.14 Métricas e Medidas.....	119

<b>5 – CONCLUSÃO.....</b>	<b>127</b>
---------------------------	------------

<b>REFERÊNCIAS.....</b>	<b>134</b>
-------------------------	------------

## **APÊNDICES**

APÊNDICE A - EVOLUÇÃO HISTÓRICA DA CONSOLIDAÇÃO DO USO DO COMPUTADOR NO CAMPO DO PROJETO (1950/2000).....	143
APÊNDICE B - ATRIBUTOS DAS QUALIDADES INTERNA E EXTERNA: NBR ISO/IEC 9126-1.....	144
APÊNDICE C - ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO.....	147
APÊNDICE D - RESULTADO EXPERIMENTO PRÁTICO COM O CADUNTUN.....	157
APÊNDICE E - RESULTADO EXPERIMENTO PRÁTICO COM O INKSCAPE.....	160
APÊNDICE F - RESULTADO EXPERIMENTO PRÁTICO COM O BLENDER.....	161
19. APÊNDICE G - CONFIGURAÇÃO DOS EQUIPAMENTOS UTILIZADOS NO EXPERIMENTO.....	165

20.	APÊNDICE H - MODELO DA FICHA DE AVALIAÇÃO.....	
	166	
21.	APÊNDICE I - QUESTIONÁRIOS DE VERIFICAÇÃO DO PERFIL DO USUÁRIO... 170	
22.	APÊNDICE J - MODELO DO QUESTIONÁRIO DE VERIFICAÇÃO DA SATISFAÇÃO DO USUÁRIO.....	177

## **INTRODUÇÃO**

Os grandes avanços alcançados no campo da informática e das telecomunicações a partir de meados da década de 1970, em consequência, principalmente, do desenvolvimento da micro eletrônica, culminaram na miniaturização dos computadores. Na medida em que se permitiu a fabricação de computadores cada vez mais compactos, aumentou-se exponencialmente o seu poder de processamento e reduziu-se o custo dos equipamentos. A disseminação gradativa do uso destas máquinas nos meios corporativos, aliado às transformações econômicas e sociais desencadeadas pelo processo de globalização dos mercados, acarretou a informatização em diversas atividades em substituição a processos manuais.

Os efeitos desta automação vêm-se sentindo com crescente rapidez, onde profissionais e empresas são pressionados pela necessidade de alinharem-se aos conceitos de economia e racionalidade da produção em resposta às exigências do mercado globalizado. Como reflexo disto, empresas e profissionais de arquitetura são induzidos a se inserirem no contexto da automação, pressionados pela necessidade de desenvolverem produtos, projetos e desenhos de melhor qualidade, estudos alternativos e elementos complementares, em menos tempo e com menor custo (AMORIM, 2000). O processo projetual em arquitetura vem gradualmente adotando ferramentas da tecnologia computacional, mais especificamente as da computação gráfica, como instrumentos não só de representação, mas como um ambiente integrado de projeto, incorporando simulações e outros recursos que dificilmente seriam possíveis pelos meios tradicionais.

Ao mesmo tempo, a evolução tecnológica, a modificação do processo de produção e as crescentes necessidades da sociedade globalizada, levaram à introdução de disciplinas que tratam da automação do projeto em diversas universidades dedicadas ao ensino da arquitetura e urbanismo, como forma de reestruturação e modernização da grade curricular, com a finalidade de orientar os futuros profissionais para a realidade de um mercado cada vez mais exigente e competitivo.

O trabalho que se segue visa contribuir para o estudo de alternativas de acesso às tecnologias de automação do projeto, abordando as perspectivas de utilização de *softwares* gráficos livres<sup>1</sup> na prática projetual em arquitetura, observadas a atualidade e urgência do debate em torno da questão.

---

<sup>1</sup> Refere-se ao *software* de código aberto, produzido de forma colaborativa cuja licença de uso permite copiar, modificar e redistribuir o mesmo gratuitamente, garantindo também o acesso ao código fonte do programa.

Contudo, sendo este um tema pouco explorado, emergente de discussões recentes, surgidas em grupos especializados que contestam de forma mais ampla a apropriação do conhecimento e o estabelecimento de monopólios tecnológicos, faz-se necessário contextualizá-lo antes do seu desenvolvimento.

A transmissão e intercâmbio de conhecimentos na cultura humana fazem parte do processo de socialização, alternando-se períodos de crises e desenvolvimento, resultando no grande esquema de mundialização das idéias que hoje se experimenta. Com o decorrer do processo de formação das civilizações, determinadas culturas se sobrepuseram a outras, inicialmente pela força bruta e, mais tarde, pela força do capital. Esta última vem se aperfeiçoando, sobrevivendo aos períodos de crise ao longo do tempo e persiste até hoje, tendo como um dos principais aliados a manipulação do conhecimento e a produção de tecnologia.

O desenvolvimento alcançado pelas tecnologias computacionais nas últimas décadas, principalmente com o desenvolvimento e aperfeiçoamento contínuo do microcomputador e dos meios de transporte e de telecomunicação, têm encurtado radicalmente as distâncias, permitindo processos articulados em diferentes localidades do globo terrestre. Este é um caminho sem volta, integrante do processo acumulativo e de desenvolvimento do capital.

A automação de processos acontece de forma generalizada, como expressão de um tempo em que o capital necessita de fluidez, com base no conceito de mais valia, onde se pretende uma maior produtividade em menor espaço de tempo, aumentando, conseqüentemente, as taxas de lucro, que é o objetivo final do sistema capitalista. Assim sendo, a automação de processos passou a ser de vital importância para o desenvolvimento do capital, e daí vem o fenômeno da crescente corrida para a capacitação tecnológica. A inclusão digital passou a ser uma prerrogativa de sobrevivência na sociedade moderna e nos mercados globalizados.

No Brasil, a aplicação de *software* comercial proprietário<sup>2</sup> para qualquer fim é regida pela Lei nº 9.609/98, denominada de Lei do *Software*, que dispõe sobre a proteção da propriedade intelectual de programas de computador e sua comercialização no país, e pela Lei nº 9.610/98, que consolidou a legislação sobre direitos autorais. De acordo com essa legislação, a violação dos direitos do autor do *software*, pode acarretar em detenção de seis meses a

---

<sup>2</sup> O *software* proprietário é aquele cujo uso, redistribuição ou modificação estão proibidos, ou restritos, de tal forma que seja necessário requerer autorização aos detentores dos direitos autorais.

quatro anos, e multa que pode chegar ao montante de três mil vezes o valor original do programa, se a infração consistir na reprodução não autorizada do mesmo, independente do meio, para fins de comércio, direta ou indiretamente. Em contraponto, observa-se que a prática da “pirataria” dissemina-se nos mais diversos meios, relacionada, dentre outros fatores, com a necessidade de modernização dos processos produtivos e o alto custo para a obtenção destes recursos. Para citar o exemplo de uma ferramenta computacional aplicada na prática da arquitetura, o *software* AutoCAD da Autodesk, um dos líderes de mercado em seu segmento, na sua versão 2012, mais recente, está sendo comercializado no Brasil ao custo aproximado de U\$ 6.000,00 (seis mil dólares), segundo cotação feita em revenda autorizada da Autodesk, sediada na cidade de Salvador - Bahia, sendo que esse valor é equivalente à licença de uso do programa restringida a um único computador, ou seja, se faz necessário adquirir uma cópia do programa para cada máquina no caso de um escritório de arquitetura, e uma vez que o profissional, ou a firma, queira manter-se atualizado quando do lançamento de novas versões do programa, cuja frequência geralmente é anual, terá que pagar, aproximadamente, 50% do valor da versão adquirida originalmente, por máquina a ser atualizada.

Ocorre atualmente um grande debate a nível mundial em torno do tema da inclusão digital, cuja base é o desenvolvimento e a produção de tecnologias acessíveis a todas as camadas sociais, haja visto que o processo de desenvolvimento da cultura humana leva à necessidade de se adotar novos procedimentos calcados no uso da informática em vários níveis, desde as mais triviais até as mais especializadas tarefas, sob pena de distanciamento do processo evolutivo e conseqüente exclusão social. É sob esta ótica, atento aos mecanismos capitalistas que determinam quanto uma pessoa tem que pagar pela utilização de um programa de computador, e os instrumentos legais colocados à disposição dos exploradores deste tipo de mercado, que surge a filosofia do *software* livre, cujo discurso contesta os mercados proprietários mais poderosos da indústria do *software* e tem chamado a atenção de grupos que são contrários à apropriação restritiva do conhecimento; os que vêem nesta filosofia uma oportunidade de derrubar os gigantes da indústria de *software*, até corporações que enxergam nos programas livres uma oportunidade de se desfazer de uma incômoda taxa de monopólio que restringe seus negócios (STEFANUTO; SALLES, 2005).

Os canais de comunicação abertos pela internet propiciaram a organização destes grupos de forma geograficamente descentralizada, ampliando gradativamente o número de comunidades

virtuais de programadores que praticam diferentes formas de organização para o desenvolvimento colaborativo de *software* livre para uma gama diferenciada de aplicações. Dentro deste contexto, o presente estudo segue rumo à investigação de programas gráficos livres, esperando alcançar resultados que contribuam para cobrir lacunas importantes de informação referente às questões do acesso às tecnologias computacionais e às ferramentas CAD aplicadas à prática do projeto de arquitetura e, ao mesmo tempo, servir de referência para o desenvolvimento de novos estudos que abordem esta mesma temática.

### **Pressupostos, objetivos e metodologia**

A abordagem deste trabalho encontra justificativa para o seu desenvolvimento a partir dos seguintes pressupostos:

1. A corrida para a capacitação tecnológica, diante das exigências impostas pelo mercado globalizado, pressiona empresas e profissionais de arquitetura a se inserirem no contexto da automação. No entanto, as ferramentas computadorizadas de auxílio ao projeto mais difundidas e empregadas são provenientes de empresas desenvolvedoras de *software* com finalidades exclusivamente comerciais, que cobram preços altos e, muitas vezes, restritivos pela licença de utilização destes programas, expondo os usuários em situação irregular a graves sanções legais, atingindo, principalmente, pequenas corporações, os setores público e acadêmico, bem como usuários autônomos.
2. O debate em torno do tema da inclusão digital tomou proporção mundial, suscitado pela urgência na busca de alternativas de acesso às tecnologias computacionais aplicadas nos mais diversos níveis das atividades humanas. No campo da arquitetura, a automação do projeto se estabelece de forma sólida tanto no ensino como na prática profissional, visando a inserção do seu contingente produtivo na realidade de um mercado cada vez mais exigente e competitivo. Entretanto, existe uma grande lacuna referente a estudos que apontem alternativas direcionadas à ampliação do acesso às ferramentas da tecnologia CAD<sup>3</sup>.

---

<sup>3</sup> CAD (*Computer Aided Design*) Projeto Auxiliado por Computador, traduzido para o português. Termo genérico adotado para englobar todos os tipos de programas específicos para uso em projeto.



3. A inclusão de disciplinas voltadas para a informatização do processo projetual na grade curricular de diversas Faculdades de Arquitetura e Urbanismo tem, frequentemente, levado para as salas de aula matérias que adotam quase que exclusivamente programas comerciais proprietários como ferramentas para aplicações no campo do projeto, constituindo, dessa forma, uma escolha restritiva, na medida em que despreza ou mesmo desconhece outras alternativas, e acaba por alimentar um ciclo vicioso com as práticas do mercado de trabalho, fortalecendo o estabelecimento do monopólio de determinadas marcas e patentes. Na medida do possível, é preciso difundir conceitos de desenvolvimento computadorizado do projeto, sem se ater a programas restritivos, oferecendo alternativas de acesso a essas tecnologias.
  
4. A característica independente e não uniforme da produção e da divulgação do *software* livre, ao mesmo tempo em que ele se estabelece como alternativa promissora de acesso à informatização em diversos campos do conhecimento, leva a indagar sobre a existência e possibilidade de aplicação, de ferramentas desta natureza no campo do projeto de arquitetura, face aos pressupostos anteriormente enumerados.

Nesse sentido, o estudo que se segue tem como objetivo geral identificar programas gráficos livres e avaliar o potencial de aplicação dessas ferramentas no desenvolvimento de projetos de arquitetura. De forma específica, objetiva:

- Identificar, classificar e selecionar programas enquadrados nessa categoria;
- Analisar o atual estágio de desenvolvimento e difusão dessas ferramentas;
- Analisar os recursos disponíveis e as potencialidades de aplicação dos programas selecionados;
- Propor uma forma de avaliação qualitativa de *softwares* livres com potencial de utilização na arquitetura;
- Configurar o panorama atual de aplicação dos *softwares* gráficos livres, e tecer recomendações referentes à utilização e desenvolvimento dos mesmos para atenderem às expectativas da prática projetual em arquitetura.

Os procedimentos metodológicos empregados apoiam-se em três pontos básicos:

- Utiliza referências bibliográficas para tratar aspectos históricos, filosóficos e conceituais do *software* livre de um modo geral, a questão da trajetória da automação do projeto ao longo do tempo e o estado da arte das ferramentas CAD comerciais, com o intuito de estabelecer um paralelo entre as formas livre e proprietária de produção desta tecnologia. Utiliza também esse tipo de referência para apropriação de conhecimento sobre avaliação de *software* e fundamentação científica sobre metodologia da pesquisa qualitativa;
- Utiliza prescrições normativas da ABNT (Associação Brasileira de Normas Técnicas) originadas a partir da ISO/IEC (*International Organization for Standardization/International Electrotechnical Commission*) para definição dos procedimentos e técnicas utilizadas no processo de seleção e avaliação dos *softwares*;
- Utiliza a internet como fonte de pesquisa para identificação e seleção dos programas gráficos livres e das comunidades de usuários e desenvolvedores;
- Por fim, são realizadas experiências práticas de uso destas ferramentas, ressaltando-se o ponto de vista do usuário no que se refere ao objeto do estudo e aos resultados alcançados em uma avaliação de abordagem qualitativa.

Ao final desse processo busca-se o estabelecimento de relações entre os resultados quantitativos e qualitativos conduzindo, assim, a conclusões e recomendações a respeito do desenvolvimento e da utilização destes programas na prática projetual de arquitetura.

A abordagem adotada resultou na estruturação do trabalho da seguinte forma:

**Software Livre: uma alternativa de inclusão tecnológica** – inicialmente trata da gênese do *software* livre, sua origem e os caminhos percorridos pelos seus criadores e seguidores para implementar um modelo de desenvolvimento e de negócio com esta “nova” modalidade de *software*, traçando um breve histórico da trajetória do projeto GNU/Linux, principal exemplo de sucesso nesta jornada. Em seguida, traz ao entendimento aspectos conceituais que definem um programa como inserido dentro desta categoria, ao tempo em que discute a ideologia que norteou o seu desenvolvimento. Em contraponto, prossegue com um debate sobre aspectos

relacionados à propriedade intelectual e ao direito de uso de programas de computador. Para finalizar, traça um panorama mais amplo do atual estágio de desenvolvimento destas ferramentas, sob a ótica de sua produção e consumo, abordando a forma de organização e o perfil das comunidades de programadores e usuários, o modelo de desenvolvimento, dispositivos de controle de qualidade, áreas mais frequentes de aplicação dos programas livres e o panorama de inserção dessas ferramentas no contexto nacional.

**Evolução da Computação e Programas Gráficos** – acompanha o desenvolvimento evolutivo do aparato computacional empregado na atividade projetual, desde o surgimento do primeiro computador eletrônico digital até a aparição dos primeiros sistemas CAD, abordando os avanços tecnológicos que viabilizaram a miniaturização dos computadores e o desenvolvimento da computação gráfica, e sua conseqüente disseminação como ferramenta de auxílio ao projeto. Para configurar o cenário atual de desenvolvimento da tecnologia computacional aplicada ao projeto, uma segunda abordagem analisa o estado da arte e suas tendências de desenvolvimento, com base na literatura existente. Em seguida, dado o tema deste trabalho, abre-se uma discussão a respeito dos tipos de ferramentas CAD mais difundidos, as tecnologias a elas integradas e, de uma forma generalizada, como elas se aplicam à projeção, descendo a recortes conceituais que permitirão uma classificação de aplicações mais específicas.

**Avaliação de Software: Programas livres gráficos utilizados na Arquitetura** – Esse capítulo explora a questão do *software* livre gráfico com potencial de aplicação na produção de projetos de arquitetura. Com o auxílio da internet, buscou-se identificar estes programas na página *web sourceforge.net* – site que hospeda milhares de projetos de *software* livre para diversas finalidades - e reunir três ferramentas com características potenciais de aplicação ao processo de projeto, para, posteriormente, submetê-las a um processo de avaliação. Para seleção dos programas e o planejamento do processo de avaliação utilizou-se métodos e técnicas prescritos pela ABNT (Associação Brasileira de Normas Técnicas), referente à avaliação de produto de *software* e literatura especializada da área de Ciência da Computação, dentro dessa temática.

**Uma experiência de avaliação em uso** – Nesse capítulo, aplica-se a metodologia de avaliação, seguindo as etapas sintetizadas no capítulo anterior, dando origem ao plano de avaliação das ferramentas selecionadas e posterior execução do experimento, durante o qual

segue-se uma abordagem baseada nos registros realizados durante seu desenvolvimento, focados no acompanhamento e observação de um pequeno grupo de usuários, suas impressões e reações à aplicação dos programas na elaboração de um projeto de arquitetura de baixa complexidade, onde a questão estatística, relacionada com o tamanho da amostragem não apresenta relevância para fins do estudo, cujo viés adotado fundamenta-se nas características, usos e possibilidades da pesquisa qualitativa (FERREIRA; CALVOSO; GONZALEZ, 2002; NEVES, 1996). Com base nos registros das ocorrências e dos resultados alcançados, segue-se uma análise interpretativa, relacionando dados quantitativos e qualitativos, que subsidiará a formulação de conclusões e recomendações.

Na **conclusão**, procura-se contextualizar os resultados obtidos com o processo de avaliação dos programas livres gráficos e tecer formulações sobre o seu estabelecimento como alternativa de acesso à tecnologia computadorizada aplicada ao projeto. Num segundo momento são feitas proposições para o incremento no desenvolvimento destas ferramentas e elaboração de novos estudos dentro desta mesma temática, ainda pouco explorada.

**1. SOFTWARE LIVRE:  
uma alternativa de inclusão tecnológica**

## 1.1 Breve Histórico

A idéia do desenvolvimento colaborativo de *software* nasceu com o próprio computador. Neste tempo, redistribuição e aprimoramento de programas eram comuns e vistos de forma positiva. Para Augusto (2003), esse processo colaborativo é um dos responsáveis pela formulação de boa parte dos conhecimentos que formam o arcabouço teórico do que se conhece hoje como Ciência da Computação. Entretanto, apenas nos últimos anos, e devido principalmente ao sucesso do sistema operacional Linux, o tema *software* livre ganhou interesse maior da comunidade científica e de usuários de ferramentas computacionais.

Em 1971, Richard Stallman<sup>4</sup> iniciou suas atividades no Laboratório de IA<sup>5</sup> do MIT<sup>6</sup>, incorporando-se a uma comunidade de programadores que compartilhavam *software* há vários anos (STALLMAN, c1998). O trabalho que Stallman desenvolvia no Laboratório estava relacionado com o aprimoramento do sistema ITS<sup>7</sup> (*Incompatible Timesharing System*) desenvolvido pela equipe do MIT para o PDP-10, um dos computadores de grande porte na época. Segundo Levy (1994), o ITS foi um dos sistemas mais importantes para o desenvolvimento da cultura *haker*<sup>8</sup> pelo fato de que, diferente dos sistemas de tempo compartilhado até então existentes, permitia que um mesmo usuário pudesse executar vários programas ao mesmo tempo e não utilizava senha para acesso possibilitando, assim, que qualquer um pudesse abrir arquivos pessoais armazenados no computador e olhar quais programas estavam sendo desenvolvidos por outros *hackers*, procurar por *bugs*<sup>9</sup> e consertá-los (LEVY, 1994). Àquele tempo não existia o conceito de *software* livre, mas a intensa troca de informações e programas que se dava entre as comunidades de programadores caracterizava uma prática de livre distribuição, modificação e, até mesmo, à “canibalização”<sup>10</sup> de parte de

<sup>4</sup> Richard Stallman, americano, físico pela Universidade de *Harvard* em 1974 e criador da Fundação *Software Livre*.

<sup>5</sup> A Inteligência Artificial (IA) é uma importante área de pesquisa da Ciência da Computação dedicada a buscar métodos ou dispositivos computacionais que possuam ou simulem a capacidade humana de resolver problemas, pensar ou, de forma ampla, ser inteligente. Fonte: <[http://pt.wikipedia.org/wiki/Intelig%C3%A2ncia\\_artificial](http://pt.wikipedia.org/wiki/Intelig%C3%A2ncia_artificial)>. Acesso em: 05 fev. 2007.

<sup>6</sup> O Instituto Tecnológico de Massachusetts, em inglês *Massachusetts Institute of Technology* (MIT), é um centro universitário de educação e pesquisa privado localizado em Cambridge, Massachusetts, nos EUA. Em português: Sistema Incompatível de Tempo Compartilhado.

<sup>8</sup> Hacker é o termo utilizado para denominar indivíduos com conhecimento avançado na elaboração e modificação de *software* e *hardware*, seja desenvolvendo utilidades novas ou melhorando/adaptando antigas. Muitas vezes aplica-se esse termo de forma equivocada para denominar indivíduos que invadem sistemas computacionais com o intuito de violá-los, os quais denomina-se *Cracker*.

<sup>9</sup> Bug é um erro no funcionamento comum de um *software*, também chamado de falha na lógica programacional, pode causar discrepância no objetivo, ou impossibilidade de realização, de uma ação na utilização de um programa de computador.

<sup>10</sup> O termo canibalização em computação refere-se à incorporação do código fonte de um determinado

alguns programas para constituir novos produtos.

No início dos anos de 1980, a comunidade de programadores do laboratório de IA do MIT se dissolveu em consequência das novas regras impostas pelo mercado que passou a lançar computadores que já vinham com o próprio sistema operacional, de “arquitetura fechada”. Nesse período, muitos dos programadores do laboratório do MIT foram contratados por empresas privadas, mas Stallman recusou-se a se associar ao mundo corporativo de exploração desse “novo” nicho mercadológico para iniciar um projeto que seria capaz de, outra vez, tornar possível a recomposição de uma comunidade de programadores em torno do desenvolvimento de *software* de forma colaborativa e aberta (STALLMAN, c1998). Para seu empreendimento, ele dedicou-se ao desenvolvimento de um sistema operacional compatível com o Unix<sup>11</sup>, que já era bem difundido no mercado de informática, ao qual denominou de Projeto GNU, como um acrônimo recursivo para *GNU's Not Unix*, querendo, com isso, transmitir a idéia de que o seu sistema seria compatível com o Unix, e não uma nova versão dele (STALLMAN, 1985). O desenvolvimento de um sistema complexo como este representava um grande desafio e iria requerer muito tempo de trabalho. Para racionalizar esta tarefa, Stallman resolveu que utilizaria, sempre que possível, outros “*softwares* livres” já desenvolvidos, agregando-os ao conjunto de ferramentas que comporia o seu sistema.

A viabilização do uso da Internet permitiu que o projeto de Stallman ganhasse divulgação e contasse com uma comunidade de programadores simpatizantes, impulsionando o desenvolvimento dos programas que comporiam o Sistema e, ao mesmo tempo, permitindo que esses programas pudessem ser disponibilizados na rede.

Em setembro de 1984, Stallman começou a trabalhar no GNU Emacs<sup>12</sup>, que nos primeiros meses de 1985 já possuía funcionalidades executáveis. Com a divulgação dessa ferramenta em meios especializados, surgiram usuários interessados em utilizá-lo. Mas para evitar que pessoas ou instituições que tivessem acesso ao *software* distribuído viessem a se apropriar do produto modificado a partir da utilização do código-fonte, inferindo-lhe uma licença restritiva, tal qual a dos *softwares* proprietários, Stallman utilizou um mecanismo legal ao qual

---

programa, ou parte dele, na composição do código fonte de outros programas.

<sup>11</sup> UNIX é um sistema operacional portátil (ou portátil), multitarefa e multiusuário) originalmente criado por um grupo de programadores da AT&T e dos Bell Labs, que incluem Ken Thompson, Dennis Ritchie e Douglas McIlroy.

<sup>12</sup> O Emacs é um editor de texto, usado notadamente por programadores e usuários que necessitam desenvolver documentos técnicos, em diversos sistemas operacionais.

denominou de *copyleft*<sup>13</sup> dando origem à licença GNU GPL<sup>14</sup>, que garantia ao produto GNU a característica de perpetuação da liberdade de uso, modificação e/ou redistribuição (STALLMAN, c1998).

Com a crescente procura pelo Emacs, e o ingresso de novos membros ao projeto GNU, ocorreu a necessidade de criação de uma estrutura formal para angariar fundos e dar continuidade ao trabalho de produção de *software* livre. Assim, em 1985, Stallman cria a FSF (*Free Software Foundation*)<sup>15</sup>, uma organização sem fins lucrativos, que aceita doações, mas tem a sua principal receita provinda da venda de cópias de *software* livre e outros serviços relacionados.

Nesse período, a comunidade já havia avançado bastante no desenvolvimento do sistema operacional GNU, mas o *Kernel*<sup>16</sup> do sistema ainda não estava pronto. Em 1991, Linus Torvalds, estudante de graduação da Universidade da Finlândia, anuncia o lançamento de um *Kernel* compatível com o Unix, desenvolvido por ele, ao mesmo tempo em que convoca programadores dispostos a “ressucitar” a comunidade de *hackers*, no sentido de trabalharem em conjunto, para dar continuidade ao desenvolvimento de seu programa, ao qual denominou de Linux (AUGUSTO, 2003). Em 1992, Torvalds adotou a licença de distribuição GPL para o seu produto e começou a integrar os elementos GNU já existentes no seu *Kernel*. Ao combinar o Linux com o sistema GNU deu-se origem a um sistema operacional completo, o sistema GNU/Linux.

Com o aperfeiçoamento gradual do Linux, surgiram projetos que agregavam um conjunto de aplicações pré-configuradas, integradas ao sistema operacional, com um sistema de instalação simplificado e uma marca associada às distribuições Linux, concomitantemente à criação de estruturas organizacionais para a comercialização desses produtos, financiamento do trabalho dos desenvolvedores e garantia da qualidade, já que cada marca é responsável pelo pacote que

---

<sup>13</sup> *Copyleft* é um trocadilho com o termo *copyright*. Traduzido literalmente para o português não faz sentido, mas segundo a FSF significa “deixamos copiar”. (FREE SOFTWARE FOUNDATION, c1996).

<sup>14</sup> GNU *General Public License* (Licença Pública Geral), GNU GPL ou simplesmente GPL, é a designação da licença para *software* livre idealizada por Richard Stallman no final da década de 1980, no âmbito do projeto GNU da *Free Software Foundation*. Esta licença é que implementa os termos específicos de distribuição que compõem o *copyleft*.

<sup>15</sup> Em português: Fundação *Software* Livre.

<sup>16</sup> *Kernel* de um sistema operacional é entendido como o núcleo deste ou, numa tradução literal, cerne. Ele representa a camada do *software* mais próxima do *hardware*, sendo responsável pela interface para utilização dos recursos do sistema computacional como um todo.



distribui (REIS, 2003).

O sucesso aparente alcançado pela comunidade Linux e o projeto GNU despertou o interesse de programadores em produzir *software* livre como alternativa para o desenvolvimento de pacotes para fins específicos, convertendo gradualmente aplicativos disponíveis para a plataforma Unix comercial para a plataforma GNU/Linux e, assim, começou-se a difundir com maior amplitude projetos em torno do *software* livre. Com o passar do tempo, e dado o isolamento geográfico dos envolvidos com esses projetos, foram sendo criadas *websites* e listas de correio eletrônico (FAQ)<sup>17</sup> dedicados à discussão e veiculação de notícias de interesse de desenvolvedores e usuários desses programas, estabelecendo assim fatores importantes de consolidação e formação de uma consciência real de comunidade (REIS, 2003).

Hoje, o GNU/Linux, ou simplesmente Linux, como é mais conhecido, é o principal exemplo de sucesso em desenvolvimento e modelo de negócio em *software* livre. Através da FSF ele ganhou uma gama variada de distribuições, abrindo caminho para o surgimento de diferentes alternativas de aplicações em processos produtivos, educacionais e organizacionais. A comunidade de *software* livre é cada vez mais ampla e seus produtos cada vez mais confiáveis e diversificados, sendo objeto de teses e dissertações preocupadas em discutir os rumos do acesso às tecnologias computacionais, indispensáveis ao mundo contemporâneo globalizado, a exemplo dos diversos títulos constantes das referências bibliográficas consultados para o desenvolvimento desse estudo.

Ao longo dos anos, a ambiguidade do termo *software* livre, como sinônimo de *software* grátis - cujas diferenças serão exploradas de forma detalhada mais adiante no item 1.2 - foi alvo de várias críticas dentro e fora do movimento, colocando em dúvida sua qualidade e requisitos de eficiência, afastando a expectativa de adoção destes programas no meio corporativo, associando-os a estereótipos negativos. No início do ano de 1997, um grupo de líderes da comunidade de programadores, entre eles Eric Raymond<sup>18</sup>, Bruce Perens<sup>19</sup> e Tim O'Reilly<sup>20</sup>,

---

<sup>17</sup> FAQ é um acrônimo da expressão inglesa *Frequently Asked Questions*, que significa Perguntas Frequentes. Num sentido amplo, significa uma compilação de perguntas frequentes acerca de determinado tema.

<sup>18</sup> Eric Steven Raymond, também conhecido como ESR, é um famoso hacker americano. Autor de *A Cathedral e O Bazar*. É quem mantém o *Jargon File*, mais conhecido como *The Hacker's Dictionary* (O Dicionário dos Hackers).

<sup>19</sup> Bruce Perens foi líder do projeto Debian, uma das mais famosas e bem sucedidas distribuições Linux, durante o período de abril de 1996 a dezembro de 1997, fundador dos projetos *Linux Standard Base e User*

frustrados com a falta de reconhecimento do potencial do *software* livre como fonte de inovações e como base de desenvolvimento do *software* comercial, mesmo havendo exemplos de adoção comercial destes produtos, resolveram atacar o problema através de uma campanha de *marketing* com o objetivo de realçar as características de confiabilidade, custo e redução do risco estratégico do negócio apoiado pela tecnologia do *software* livre (AUGUSTO, 2003).

A primeira decisão do grupo foi mudar a terminologia que define o desenvolvimento do *software* feito de forma colaborativa, o termo *software* livre (*free software*) deu lugar ao termo (*Open Source software*)<sup>21</sup>, evitando as conotações negativas que o primeiro termo suscitava. Assim o *Open Source* definiu um tipo de marca, para o qual foi criada a OSD (*Open Source Definition*)<sup>22</sup>, objetivando esclarecer os requisitos necessários para que um *software* seja classificado como livre e empregá-lo a diversos tipos de licença de distribuição, menos ou mais restritivas. Entretanto, para Stallman, o termo *software* livre define melhor o movimento, dando ênfase a aspectos sociais e questões filosóficas sobre sua utilização e desenvolvimento, enquanto *Open Source* define apenas uma metodologia de desenvolvimento.

Segundo Augusto (2003), com a evolução e a facilidade de acesso à Internet, os projetos de *software* livre iniciados por autores individuais, ou grupos pequenos, começaram a atrair a atenção de grupos maiores de desenvolvedores ativos, que gradualmente consolidaram-se em organizações informais para o aprimoramento destes produtos. Assim, formou-se uma série de comunidades em torno de programas de código aberto, com participação espontânea de usuários e programadores. O contexto atual é o de milhares de projetos independentes, para as mais diversas finalidades e campos de atuação. É neste contexto que este estudo objetiva identificar *softwares* livres gráficos e avaliar seu potencial de uso na prática de projetos de arquitetura, temas esses desenvolvidos nos capítulos 3 e 4.

## 1.2 Aspectos Conceituais e Filosóficos do *Software* Livre

Partindo do princípio de que a informatização dos processos tornou-se uma das premissas

---

*Linux*, e cofundador da *Open Source Initiative*. É autor da série de livros *Bruce Perens' Open Source Series*, publicados por Prentice Hall.

<sup>20</sup> Tim O'Reilly é o fundador da O'Reilly Media, editora americana de livros, desenvolvedora de *sites web* e organizadora de congressos. Faz parte da diretoria da CollabNet, e foi da diretoria da Macromedia até sua fusão com a Adobe em 2005.

<sup>21</sup> Em português: *Software* de código aberto.

<sup>22</sup> Em português: Definição de código aberto.

para o desenvolvimento das sociedades modernas, e de que o compartilhamento do conhecimento está diretamente relacionado com o desenvolvimento acelerado das tecnologias como um todo, e em particular da tecnologia computacional, a comunidade do *software* livre difunde uma filosofia calcada na construção de alternativas para a inclusão digital das camadas sociais menos favorecidas e combate às restrições impostas pelo mercado de *software*, acreditando que tais restrições contribuem para a redução do bem estar que determinados segmentos da sociedade pode usufruir pelo uso de um programa de computador.

No entanto, a Free Software Foundation (c1996a) esclarece que não se trata de uma questão meramente econômica, mesmo porque o termo *software* livre não se refere exclusivamente a um programa gratuito. A palavra livre pode ter dois significados; tanto pode estar se referindo a grátis como pode estar se referindo à liberdade, no sentido mais estrito da palavra. Algumas vezes os programas livres são distribuídos gratuitamente, outras vezes se cobra por eles, contudo não quer dizer que um preço baixo ou a gratuidade torne um programa mais livre do que outro (FREE SOFTWARE FOUNDATION, c1996c). Mais precisamente, ele se refere a quatro tipos de liberdade:

5. A liberdade de executar o programa, para qualquer propósito;
6. A liberdade de estudar como o programa funciona, e adaptá-lo conforme as necessidades. Acesso ao código-fonte é um pré-requisito para esta liberdade;
7. A liberdade de redistribuir cópias;
8. A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos. Essa liberdade também depende de acesso ao código-fonte (FREE SOFTWARE FOUNDATION, c1996a).

Um programa é *software* livre se os usuários possuem essas quatro liberdades, sem estarem sujeitos a penalidades. Seus defensores alegam que não se trata de uma questão de combate ao sistema de comercialização e “*copyright*”<sup>23</sup>, mas, sim, de combate à restrição do conhecimento, pois acreditam não haver nada de errado em se vender um *software*, contudo, consideram existir distorções na forma em que é usualmente comercializado, pois defendem que o valor de um programa de computador deve estar atrelado à contribuição social que ele

---

<sup>23</sup> Traduzido literalmente para o português significa: direito de cópia. Mais conhecido como direito autoral é a denominação usualmente utilizada em referência ao rol de direitos legais outorgados aos autores de obras intelectuais (literárias, artísticas ou científicas).

pode trazer, e esta contribuição só se concretiza na medida em que a sociedade pode usufruir plenamente dos seus resultados (FREE SOFTWARE FOUNDATION, c1996b). Defendem, ainda, a não existência de direito intrínseco sobre a propriedade intelectual, e que os tipos dessa suposta propriedade, que o Estado reconhece, foram assim estabelecidas para encorajar os inventores a divulgarem os detalhes de suas invenções, conseqüentemente ajudando a sociedade no desenvolvimento do conhecimento (STALLMAN, 1992).

No artigo intitulado “Porque o Software Deveria Ser Livre”, Stallman (1992) critica o lucro excessivo gerado com o modelo capitalista de comercialização de programas de computador, porque entende que este modelo ignora o prejuízo que as restrições impostas pelas leis de proteção aos detentores de direitos autorais causam ao avanço do conhecimento, e que este prejuízo é muito maior do que o benefício que o uso restrito de programas de ampla utilidade pode gerar para a sociedade como um todo. Essa é a hipótese que Stallman levanta para sugerir que, uma vez pago o custo de desenvolvimento de um *software*, incluindo os recursos humanos, este deveria passar a ser um produto de domínio público, para que pudesse derivar o desenvolvimento do conhecimento e ter os seus benefícios ampliados. Desta forma é que desenvolvedores e usuários de *software* livre tentam mostrar que a contribuição total de um programa para a sociedade é reduzida atribuindo-se um proprietário a ele, ou, em suma, retirando-se as liberdades dos usuários.

Estes são os princípios que norteiam o movimento do *software* livre. Polêmicos, porque vão em direção contrária aos paradigmas da economia capitalista, envolvendo questões éticas e morais, mas que têm encontrado respaldo no apelo pela construção de formas colaborativas de produção e de economias solidárias, emergentes da necessidade de se reduzir o processo acentuado de exclusão vivido nas sociedades modernas.

### **1.3 O Desenvolvimento e o Acesso ao *Software* Livre**

A sistemática de desenvolvimento do *software* livre está calcada no trabalho colaborativo de redes de programadores e usuários geograficamente dispersos. A internet é o principal meio de integração e expansão desta forma de organização de trabalho, através de recursos amplamente disponíveis, de forma que todos os que podem e desejam colaborar têm como fazê-lo de diversas formas (programação, indicação de falhas, sugestão de melhorias, tradução, documentação, divulgação, projeto gráfico de páginas *web*, ou mesmo

financeiramente). A esses coletivos tradicionalmente denomina-se comunidade de desenvolvimento de *software* (SOCIEDADE SOFTEX, 2005). Em muitos casos, esses grupos são liderados por um conjunto pequeno de pessoas e, em outros casos mais raros, existe uma organização estabelecida gradualmente, consolidada de forma institucional (REIS, 2003).

De acordo com Sociedade Softex (2005), o termo “comunidade” utilizado anteriormente possui dimensões variadas. De uma forma generalista, aplica-se a todo tipo de pessoa ou grupo envolvido com o *software* livre, vinculados, de alguma forma, ao seu desenvolvimento, uso, difusão ou apoio. De forma específica, se define como comunidade aquela que se forma em torno do projeto de desenvolvimento de uma ferramenta em particular, e de uma forma mais específica ainda, como sub-comunidades que tratam de aspectos específicos dentro do projeto de desenvolvimento de determinado *software*, a exemplo de programadores (que tratam de questões técnicas sobre o desenvolvimento), e de suporte (que trata de ajuda e orientação a usuários em dificuldade para utilização de determinados recursos do programa). Os autores destacam, ainda, a existência de colaboradores esporádicos e usuários não ativos, que [...] *raramente participam dos debates sobre um software, o que os coloca em posição isolada da comunidade, mas ainda assim são considerados como tal (contam em estatísticas de usuários, por exemplo)* (SOCIEDADE SOFTEX, 2005, p. 13).

No artigo “*Evolution Patterns of Open Source Software Systems and Communities*”<sup>24</sup> Nakakoji et al, (2002), classifica os membros de uma comunidade de *software* livre em 8 níveis:

- Passivo (que é usuário mas não interage com a comunidade);
- Leitor (que assina e lê as listas de discussão do projeto);
- *Bug reporter* (que informa os defeitos);
- *Bug fixer* (que faz reparos ocasionais de defeitos);
- Desenvolvedor ocasional;
- Desenvolvedor ativo;
- Membro do núcleo.

---

<sup>24</sup> Em português: Padrões de Evolução de Sistemas e Comunidades de *Software* de Código Aberto. Disponível em: < <http://www.kid.rcast.u-tokyo.ac.jp/~kumiyo/mypapers/IWPSE2002.pdf>>. Acesso em: 18 mai. 2008.

O nível de participação de membros da comunidade varia de acordo com a motivação e o nível de interesse de cada um. Reis (2003) chama atenção para o fato de que esta divisão não apresenta rigidez, ocorrendo migrações de um papel para outro, conforme escolha e oportunidade pessoal. Acrescentar-se-ia a esta observação o nível de conhecimento do usuário/desenvolvedor a respeito das técnicas e linguagem de programação, bem como da matéria de que trata o *software*.

Augusto (2003) apresenta o quadro 1, que oferece um resumo dos principais papéis desempenhados pelos membros da comunidade do *software* livre, aos quais ele chama de *stakeholders*.<sup>25</sup>

**Quadro 1:** *Stakeholders* e seus papéis.

<b>Papel</b>	<b>Como Clientes</b>	<b>Como Atores</b>	<b>Como Donos</b>
<b>Grupo</b>			
Desenvolvedores	Usam o <i>software</i> livre como suporte ao desenvolvimento	Principais implementadores de alterações nos sistemas	Dão o direcionamento inicial do sistema, mas normalmente não têm o poder de terminá-lo.
Usuários	Usam os produtos de forma direta ou indireta.	Podem usar o <i>software</i> livre como uma caixa preta ou configurá-lo. Também podem ser agentes de mudança ao reportar <i>bugs</i> .	Devido às licenças, têm o mesmo direito de propriedade que o criador do <i>software</i> .
Empresas	Têm adotado o <i>software</i> livre de forma entusiástica.	Atuam como implementadores ou patronos das mudanças.	Têm controle sobre a marca da distribuição, mas não sobre o produto em si.
Organizações	Usam os produtos da mesma forma que as empresas.	Organizam o esforço de desenvolvimento.	Possuem o controle sobre o direcionamento e futuro dos projetos.

Fonte: Feller e Fitzgerald (2002 apud AUGUSTO, 2003, p. 30).<sup>26</sup>

<sup>25</sup> Embora a palavra *stakeholder* ainda não tenha uma tradução em português, em administração considera-se como um grupo de interesse mais amplo composto de empregados, fornecedores, clientes etc.

<sup>26</sup> As células em destaque representam o papel principal de cada grupo.

No estudo sobre as orientações e motivações de usuários e programadores brasileiros de *software* livre, Augusto (2003) identificou que estes indivíduos buscam principalmente aumentar seus conhecimentos em programação, ganhar reputação através de contribuições de código e ajuda a outros usuários, além de que, em alguns casos, esta atividade representa uma diversão e melhoria da empregabilidade. Ainda foram identificados indivíduos que recebem alguma forma de pagamento ou têm a obrigação de usar ou desenvolver os programas, além da expectativa da possibilidade de vender produtos ou serviços associados ao *software* livre. Vale ressaltar que apesar desses motivos, apresentados no referido estudo, estarem associados à participação de desenvolvedores brasileiros, encontra-se em Feller et al (2005) argumentos similares justificando o interesse de indivíduos pertencentes a sociedades estrangeiras em participar do desenvolvimento de *software* livre. O quadro 2, ilustra os fatores motivacionais de indivíduos e corporações que participam da comunidade.

**Quadro 2:** Motivações no desenvolvimento do *software* livre.

Nível	Micro (indivíduos)	Macro (organizações e comunidades)
Área		
Tecnológica	<ul style="list-style-type: none"> <li>• Atender necessidade específica.</li> <li>• Explorar a eficiência <i>do peer-review</i>.</li> <li>• Trabalhar com tecnologia de ponta.</li> </ul>	<ul style="list-style-type: none"> <li>• Atacar a crise do <i>software</i> (particularmente a qualidade).</li> <li>• Dividir tarefas entediantes com os usuários.</li> <li>• Promover a inovação.</li> <li>• Garantir a transparência das aplicações.</li> </ul>
Econômica	<ul style="list-style-type: none"> <li>• Ganhar benefícios futuros na carreira.</li> <li>• Melhorar técnica de programação.</li> <li>• Baixo custo de oportunidade (“nada a perder”).</li> </ul>	<ul style="list-style-type: none"> <li>• Mudança do paradigma do <i>software</i> como manufatura para um modelo de serviços.</li> <li>• Reforçar a marca.</li> <li>• Explorar receitas indiretas (produtos complementares e serviços).</li> <li>• Redução de custos - plataforma mais barata do que as alternativas proprietárias.</li> </ul>
Sóciopsicológica	<ul style="list-style-type: none"> <li>• Gratificação do ego.</li> <li>• Motivação intrínseca para programar.</li> <li>• Senso de pertencer à comunidade.</li> <li>• Altruísmo.</li> </ul>	<ul style="list-style-type: none"> <li>• Movimentos sociais precisam de um inimigo - ex. Microsoft.</li> <li>• Ideologia: <i>software</i> deve ser livre.</li> <li>• Modelo de trabalho para o futuro.</li> </ul>

O elemento essencial da construção de um *software* é o seu código-fonte. Em projetos de *software* livre o código-fonte é lançado através de uma mensagem enviada para sites específicos de divulgação desse tipo de projeto, a exemplo do <*freshmeat.net*> ou <*sourceforge.net*> e é então disponibilizado em um repositório na internet, onde fica hospedado, distribuído através de uma licença apropriada, em sua maioria a GNU/GPL. Mas as alterações na base, contida no repositório, são feitas por um grupo pequeno de pessoas, e, em muitos casos, por uma única pessoa responsável, denominada de *manteiner* ou mantenedor. Essas alterações de base ocorrem depois que a alteração sugerida passa por um processo de avaliação por outros desenvolvedores, num ciclo que se repete para cada alteração a ser introduzida.

Os serviços de hospedagem de projetos oferecem um conjunto de facilidades que viabilizam o desenvolvimento do *software*:

- Espaço para hospedagem de páginas *web* para o projeto;
- Espaço para oferecer versões lançadas do código-fonte para *download*;
- Possibilidade de criar listas de discussão própria;
- Um repositório CVS<sup>27</sup>, incluindo um sistema de visualização de código;
- Uma ferramenta de acompanhamento de defeitos;
- Controle de acesso para funções restritas (acesso de escrita ao CVS, e administração das listas de discussão, por exemplo);
- Formas de publicidade para o projeto através de concentradores de lançamentos, diretórios e serviços de busca. (REIS, 2003).

Este tipo de solução é bastante conveniente para projetos de pequeno e médio portes, oferecendo interfaces via *web* para manipulação dos serviços oferecidos.

A gerência do código-fonte normalmente é feita por uma ferramenta de controle de versões. A ferramenta mais utilizada é o CVS, que serve para armazenar e controlar versões de qualquer documento gerado durante o processo de desenvolvimento do *software*. Esta ferramenta opera

---

<sup>27</sup> O CVS, ou *Concurrent Version System* (Sistema de Versões Concorrentes) é um sistema de controle de versão que permite que se trabalhe com diversas versões de arquivos organizados em um diretório e localizados local ou remotamente, mantendo-se suas versões antigas e os *logs* de quem e quando manipulou os arquivos. O CVS pode ser baixado gratuitamente e tem o seu código aberto. Disponível em: <<http://pt.wikipedia.org/wiki/ CVS>> Acesso em: 11 mar. 2007.



armazenando cópias atualizadas dos arquivos do projeto, as diferenças entre estas e as versões anteriores, estabelecendo insumos para se auditar o que foi alterado. Detalhando o que foi dito, esse programa utiliza uma arquitetura cliente-servidor: um servidor armazena a(s) versão(ões) atuais do projeto e seu histórico, e os clientes se conectam a esse servidor para obter uma cópia completa do projeto, trabalhar nessa cópia e, então, devolver suas modificações.

As alterações promovidas por desenvolvedores diversos, depois de devolvidas ao repositório central, passam por uma ação de integração e resolução de conflitos, envolvendo um processo complexo e especializado que foge ao escopo desse trabalho. Tipicamente, cliente e servidor devem estar conectados por uma rede local de computadores, ou pela internet, mas o cliente e o servidor podem estar na mesma máquina se a configuração do CVS for feita de maneira a dar acesso a versões e histórico do projeto apenas a usuários locais. Apesar de ser muito utilizado, o CVS oferece algumas limitações e alguns projetos de *software* livre objetivam promover uma ferramenta de controle de versão para contornar estas limitações; uma das principais é a *Subversion*<sup>28</sup> (REIS, 2003).

Reis (2003) observa ainda que a maior parte das atenções está voltada para eventos e atividades realizadas após o lançamento do programa, em um processo típico de manutenção, e que a visão de desenvolvimento, proporcionada pelos projetos de *software* livre, é uma evolução constante, mas que o ciclo de vida destes projetos pode ser descontinuado a qualquer momento, sendo sustentado apenas pela motivação de sua equipe. Projetos abandonados poderão ser adotados por uma nova equipe de desenvolvedores.

De forma resumida, os elementos principais envolvidos no processo de desenvolvimento do *software* livre constituem: uma base de usuários, da qual uma parte são desenvolvedores, um repositório na internet que disponibiliza o programa, armazena o seu código fonte e aloja ferramentas apropriadas de controle de versão de *software* e rastreamentos de *bugs* enquanto listas de discussão fóruns e *chats*<sup>29</sup>, páginas *web* e outros são utilizados para o *feedback* dos usuários, dar suporte ao uso do programa, informar possíveis erros e promover discussões.

---

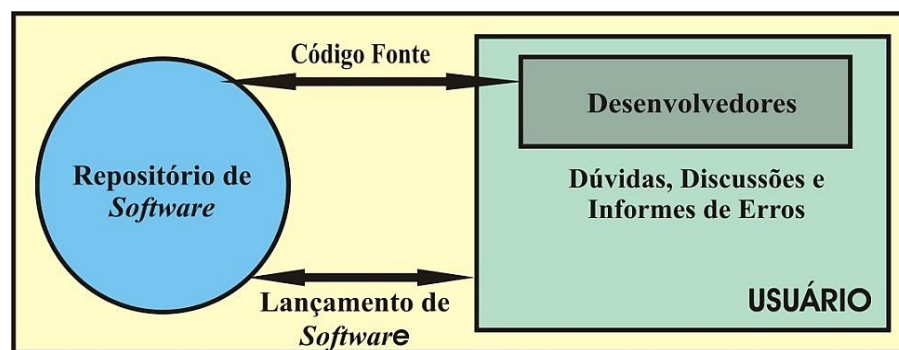
<sup>28</sup> (APACHE SOFTWARE FOUNDATION, 2011)

<sup>29</sup> Um chat, é um neologismo para designar aplicações de conversação na internet, em tempo real., a exemplo do “bate papo” utilizado no Brasil Esta definição inclui programas de IRC – *Iterative Relay Chat*, conversação em sítio *web* ou mensageiros instantâneos.

As atividades essenciais para a construção efetiva de um *software*, seguindo o modelo de produção comercial, envolvem as seguintes etapas:

- Análise e definição de requisitos - Processo de descobrir e detalhar quais as funcionalidades e limitações funcionais que o *software* requer para realizar uma determinada tarefa;
- Projeto da arquitetura - Esta atividade é um detalhamento iterativo de uma solução computacional para atender os requisitos definidos, iniciada em níveis mais abstratos e evoluindo para níveis mais refinados (projeto detalhado);
- Codificação - Consiste na criação do código-fonte, implementando a funcionalidade especificada nos requisitos, de acordo com o modelo definido no projeto;
- Teste de unidade de integração de sistema - O teste de unidade cobre partes individuais do sistema (funções, módulos e interfaces); já o teste de integração verifica o funcionamento destas partes uma vez combinados, e
- Lançamento - Uma vez o sistema (ou uma parte essencial, para processos iterativos ou cíclicos) estando pronto e validado, pode ser realizado o seu lançamento. Durante um lançamento ocorre a consolidação da documentação e ajuda *online*, o empacotamento do *software* (ordenamento do código e de informações, em pacotes, para transmissão em rede) e a sua distribuição, entrega ou instalação. É a fase onde ocorre o contato do usuário final com o sistema. (REIS, 2003).

Para ilustrar o modelo de desenvolvimento do *software* livre, Reis (2003), apresenta o diagrama da figura 1, esquematizando seus elementos fundamentais.



**Figura 1-** Diagrama de um projeto de *software* livre em desenvolvimento.

Fonte: Reis (2003, p. 15).

Segundo Reis (2003), estas atividades não apresentam uma sequência rígida, ocorrendo superposições frequentes entre elas, em uma relação sutil de interdependência que tende a ordenar a sua realização. Entretanto, a autora adverte que nos processos de desenvolvimento de *software*, sejam eles livres ou não, a fase mais importante do ciclo de vida do produto ocorre após o seu lançamento, na fase de manutenção. Além dessas atividades essenciais, atividades auxiliares são necessárias para atribuição e administração dos recursos e pessoas, do tempo de desenvolvimento, revisão e inspeção do código e à construção da documentação. No decorrer deste processo ainda se faz necessários alguns procedimentos com vistas a se garantir e manter a qualidade do artefato e do próprio processo em si.

De acordo com Associação Brasileira de Normas Técnicas (2001a) a percepção de qualidade dificilmente é obtida através da simples observação de um ou outro fator isolado, exigindo sim a aplicação de um modelo que permita avaliar a qualidade do produto de *software*, considerando sempre que possível critérios objetivos e quantitativos. O modelo SQuaRE (*Software Product Quality Requirements and Evaluation*), padronizado pela NBR ISO IEC 9126, é um dos instrumentos aplicados pela indústria para especificação e avaliação do produto de *software*. Consiste na definição de características de qualidade e seus indicadores, de forma que se possa obter uma escala de valores que indicará o grau de satisfação do usuário dentro de um determinado contexto de uso de programas de computador. A qualidade do *software*, em um modelo de produção comercial proprietário, começa a ser definida no momento de sua criação, dependendo, ao final, das medidas tomadas durante o processo, seus mecanismos de controle e especificação, tamanho e conhecimento da equipe, comunicação e articulação entre os membros que a compõem (REIS, 2003). A experiência de avaliação a ser procedida nos capítulos 3 e 4, lançará mão desses métodos e técnicas para chegar a resultados que auxiliem na verificação da qualidade em uso das ferramentas livres estudadas<sup>30</sup>.

Massey (2002), em seu artigo “*Where do Open Source Requirements come from (And what should we do about it)?*”, chama atenção para a ausência de um processo formal de obtenção de requisitos em projetos de *software* livre, começando por atender a uma necessidade pessoal de seu desenvolvedor primário, sendo incrementado por outros fatores resultantes do *feedback* de usuários. Também pode advir da implementação de padrões pré-existentes visando, principalmente, a interoperabilidade entre programas, ou ainda pela necessidade de se

---

<sup>30</sup> O conjunto de normas para avaliação de *software* utilizadas neste trabalho contam em (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2003a; 2009; 2001a; 2003b; 2003c; 2001b; 2004).

implementar protótipos com fins didáticos ou acadêmicos.

Para Healy (2003 apud REIS, 2003), a garantia de controle de qualidade irá depender da política de revisão e integração do código estabelecida pelo mantenedor, ou grupo de mantenedores do projeto. A tarefa de liderança e organização do grupo de desenvolvedores é crucial; projetos de sucesso têm líderes respeitados e tecnicamente competentes, assemelhando-se a movimentos sociais.

Dessa forma, observa-se que o modelo de desenvolvimento do *software* livre difere do utilizado pela indústria. A participação nos projetos é aberta e voluntária, sem fins lucrativos, e o código-fonte é tornado público através de licenças específicas que garantem sua utilização, modificação e posterior redistribuição sem encargo algum. Como este modelo é aberto, a constituição destes grupos pode variar em número de pessoas ao longo do processo, por diversos motivos: interesse no uso da ferramenta, disponibilidade de tempo, nível de interação com o grupo, grau de satisfação com o estágio de desenvolvimento do *software*, etc. Do ponto de vista do controle de qualidade a participação aberta pode ser ou não um aspecto positivo. Se por um lado uma grande base de usuários/desenvolvedores com acesso ao código fonte pode representar uma supervisão maior dos *bugs*, dependendo da eficácia do gerenciamento das alterações isto pode representar também uma maior vulnerabilidade a conflitos, exigindo maior responsabilidade do mantenedor, ou grupo de mantenedores da base de código. Assim, a continuidade do projeto depende do interesse e disponibilidade dos membros de um grupo que usam e desenvolvem o programa.

#### **1.4 Panorama Geral do Software Livre no Brasil**

O surgimento do *software* livre, nos mais diversos campos de atuação, aparece como uma alternativa para o desenvolvimento de sistemas acessíveis, passíveis de aperfeiçoamento e adequação às especificidades de cada ambiente ou processo produtivo, trazendo variáveis novas para as relações de mercado e comercialização destes produtos. Embora não se trate exatamente de uma ruptura tecnológica, traz uma nova forma de desenvolver e licenciar programas de computador, quebrando alguns modelos estruturais de apropriação nessa indústria.

Este modelo tem despertado grande interesse e suscitado reflexões em diferentes esferas

(governo, empresas, academia etc.) em nível nacional e internacional, pelo fato de representar uma opção de ampliação do acesso ao desenvolvimento tecnológico a partir do estabelecimento de experiências bem sucedidas, como é o caso do sistema operacional Linux. Este é um fenômeno sem precedentes na história da indústria do *software*, havendo ainda muitos aspectos a serem estudados para o entendimento do real impacto que traz para os mais diversos campos do conhecimento e da atividade produtiva. Sabe-se, entretanto, que a motivação econômica, propiciada pela diminuição dos custos na automação de processos, é uma das principais causas da adoção destas ferramentas, ao menos nos meios corporativos.

Alguns estudos demonstram que os maiores centros de desenvolvimento e aplicação do *software* livre estão geograficamente concentrados na tríade Europa, Japão e Estados Unidos (STEFANUTO; SALLES, 2005). Entretanto, na última década, o bloco de países da América do Sul vem, gradativamente, adotando políticas direcionadas para a utilização maciça de programas livres na esfera administrativa governamental, apoiando agentes interessados no desenvolvimento destas ferramentas.

No caso específico do Brasil, a comunidade do *software* livre tem crescido significativamente nos últimos anos, com a criação de representações estaduais sob forma de grupos organizados, normalmente apoiados pelas universidades, principalmente Faculdades de Matemática, Ciências da Computação e seus centros de pesquisa, contando com a participação de estudantes, empresários, agentes governamentais, grupos de usuários, *hackers* e ONG's, como é o caso do PSL-BA (Projeto *Software* Livre Bahia), formado desde o ano de 2003, no Instituto de Matemática da UFBA. Grupos como esse já encontram representação em pelo menos 13 estados brasileiros (PROJETO SOFTWARE LIVRE BAHIA, c2003). Visam, a princípio, a difusão e o debate em torno do assunto como forma de promover a ampliação do campo de atuação do *software* livre e sua institucionalização a nível nacional. Usam como veículo de interação páginas *web*, rede sociais e eventos "virtuais" na internet, bem como encontros, seminários e congressos geograficamente localizados, a nível municipal, estadual, regional e nacional, a exemplos dos seguintes, confirmados, para ocorrerem no ano de 2011:

- FISLI (Festival Internacional do Software Livre): XII edição em Porto Alegre -RS (FISLI 12, 2011);
- Fórum de Tecnologia em Software Livre: IV edição em Curitiba – PR (IV FTSL, [2011]);

- Latinoware – Conferência latinoamericana de Software Livre: VIII edição em Foz do Iguaçu – PR (LATINOWARE, 2011);
- ENSL (Encontro Nordeste de Software Livre): V edição em Maceió – AL (ENSL, 2011);
- FASOL (Fórum Amazônico de Software Livre): III edição em Santarém – PA (FASOL, 2011);
- FSLDC (Fórum de Software Livre de Duque de Caxias): III edição em Duque e Caxias – RJ (FSLDC, 2011);
- ENSOL (Encontro de Software Livre da Paraíba): V edição em João Pessoa – PB (ENSOL, 2011);
- CONISLI (Congresso Internacional de Software Livre): VIII edição em São Paulo – SP (CONSILI, 2011).

Visando conter os gastos públicos, a adoção do *software* livre pelo governo federal começou a ser definida como política pública, de forma mais efetiva, a partir do ano de 2003. No mês de janeiro daquele ano o MCT (Ministério da Ciência e Tecnologia do Governo Federal do Brasil) por intermédio da FINEP (Financiadora de Estudos e Projetos), empresa pública vinculada ao citado Ministério, lançou uma chamada pública com o objetivo de acolher propostas de apoio financeiro a projetos de pesquisa aplicada e inovação tecnológica em *software* livre, para instituições de fins não lucrativos, preferencialmente em parceria com empresas interessadas na exploração econômica dos resultados do projeto. Tais projetos deviam estimular o desenvolvimento de aplicativos em *software* livres voltados, por exemplo, para: governo eletrônico; educação; saúde; geoprocessamento; segurança; comércio eletrônico ou entretenimento. Os recursos disponibilizados foram da ordem de R\$ 4.000.000,00 (quatro milhões de reais), originários do CT-INFO (Fundo Setorial de Tecnologia da Informação do MCT) e os valores dos projetos deviam estar entre R\$ 100.000,00 (cem mil reais) e R\$ 300.000,00 (trezentos mil reais), cujos itens financiáveis foram: despesas com material de consumo; *software*; instalação, aquisição, manutenção e recuperação de equipamentos; despesas acessórias com importação; serviços de terceiros (pessoas físicas ou jurídicas); passagens e diárias; obras; instalações civis e reformas em geral. Foram recebidas 267 propostas, das quais 27 foram contempladas com o apoio financeiro, na sua maioria oriundas de fundações, universidades e centros de pesquisa (BRASIL, 2003).

Em outubro de 2003, o Governo Federal instituiu, no âmbito do Comitê Executivo do Governo Eletrônico<sup>31</sup>, o CISL (Comitê de Implementação de *Software* Livre), que desde o ano de 2006 passou ao status de Comitê Permanente Executivo, coordenado pela Serpro (Serviço Federal de Processamento de Dados) (BRASIL, [200-]). No ano de 2004, o então Comitê de implementação já havia feito, através de um processo de planejamento estratégico, o relatório final que traçava diretrizes, objetivos e ações para a implantação de programas de código aberto na administração pública (INSTITUTO DE TECNOLOGIA DA INFORMAÇÃO, [200-]). Ao todo, 18 diretrizes, 12 objetivos e 29 ações prioritárias formaram o conjunto de orientações visando garantir a migração para a tecnologia livre, dentre as quais destacou-se as seguintes:

a) Diretrizes:

- Priorizar soluções, programas e serviços baseados em *software* livre que promovam a otimização de recursos e investimentos em tecnologia da informação;
- Popularizar o uso do *software* livre;
- Utilizar o *software* livre como base dos programas de inclusão digital;
- Priorizar a aquisição de *hardware* compatível às plataformas livres.

b) Objetivos:

- Efetivar o *software* livre como ferramenta corporativa padrão do governo federal;
- Conter o crescimento do legado<sup>32</sup>;
- Disseminar a cultura de *software* livre nas escolas e universidades;
- Articular a política de *software* livre a uma política de fomento à indústria.

c) Ações:

- Migrar serviços básicos de rede e planejar, estruturar e testar a migração dos demais serviços;

---

<sup>31</sup> O Comitê Executivo do Governo Eletrônico tem o objetivo de formular políticas, estabelecer diretrizes, coordenar e articular as ações de implantação do Governo Eletrônico, voltado para a prestação de serviços e informações ao cidadão.

<sup>32</sup> Foi denominado de legado, o conteúdo existente no serviço público federal baseado em *software* proprietário.

- Criar grupo de apoio e suporte para migração do legado para *software* livre;
- Desenvolver aplicativos direcionados a projetos educacionais e pedagógicos;
- Redirecionamento de fundos públicos para *software* livre (INSTITUTO DE TECNOLOGIA DA INFORMAÇÃO, [200-]).

No ano de 2004, a FINEP lançou outra chamada pública, desta vez com objetivo de selecionar propostas de desenvolvimento de aplicações inovadoras utilizando tecnologia de computação aplicada à área de governo que, preferencialmente, deveriam ser distribuídas sob termos das licenças de *software* livre. Desta vez, seis projetos foram contemplados com o apoio financeiro (BRASIL, 2004). Naquele mesmo, ano o Observatório Econômico da SOFTEX<sup>33</sup> (Sociedade para Promoção da Excelência do Software Brasileiro), que trata-se de uma Organização da Sociedade Civil de Interesse Público, e o Departamento de Política Científica e Tecnológica da UNICAMP (Universidade de Campinas - SP) iniciaram, com o apoio do MCT, uma pesquisa para aprofundar o entendimento do impacto do *software* livre e de código aberto na indústria do *software* no Brasil (STEFANUTO; SALLES, 2005). Foram estudados aspectos como abrangência de utilização, capacitação dos desenvolvedores e, particularmente, os impactos que dizem respeito às empresas de *software* (capacitação, modelos de negócios etc), cujos resultados apresentados indicaram maior disseminação do uso de programas livres em pequenas empresas, mas haviam também registros de grandes empresas que já tinham adotado este modelo para a realização de negócios, concluindo-se que houve um movimento de disseminação dessa tecnologia da periferia para o centro da indústria brasileira.

Esta mesma pesquisa ajudou a mostrar que existia concentração geográfica no desenvolvimento e uso de programas livres no âmbito nacional: 78% dos desenvolvedores individuais, 81% das empresas desenvolvedoras, 84% dos usuários individuais e 85% das empresas usuárias encontravam-se nas regiões Sul e Sudeste do Brasil. Rio Grande do Sul e São Paulo foram apontados como os dois principais focos de desenvolvimento e uso de *software* livre no País (SOCIEDADE SOFTEX, 2005).

No ano de 2005, outro importante estudo foi realizado pela Escola de Direito da Fundação

---

<sup>33</sup> Os Agentes SOFTEX prestam apoio operacional à empresas de *software*, atuando em articulação com a iniciativa privada, governos estaduais e municipais, e contam com o suporte de centros acadêmicos e instituições de fomento para atingir as metas do setor de *software* confiadas à SOFTEX (SOCIEDADE SOFTEX, [200-])



Getúlio Vargas, sob encomenda do ITI (Instituto Nacional de Tecnologia da Informação), órgão vinculado à Casa Civil da Presidência da República. Esse estudo buscou investigar, sob a ótica do direito privado, aspectos legais relacionados com a política de implementação do *software* livre, uma vez que trata-se de um produto tecnológico que se contrapõe ao estabelecimento de monopólios e patentes no campo da informática. Ressalta-se nesse estudo que a Constituição Federal inclui a criação tecnológica em seu patrimônio cultural, e como tal, a Lei deve incentivar a produção desses bens e o conhecimento dessa produção estabelecendo mecanismos, apontando e induzindo comportamentos para que o objetivo constitucional da autonomia do conhecimento informático seja alcançado, dessa forma, atribuiu relevância ao [...] *importante dispositivo legal* [...] (FALCÃO et al, 2005, p. 108) dentro da política de informática que proíbe a criação de situações monopolísticas, de direito ou de fato, observando que isso não significa reprimir monopólios, mas impedir o controle da informação por algumas corporações. Com base nesse argumento, concluiu-se que a implementação da política do *software* livre faz oposição ao controle do conhecimento por monopólios privados e, dessa forma, encontra suporte legal na medida em que integra e dissemina o conhecimento sobre programas de computador (FALCÃO et al, 2005).

Nesse contexto, a política nacional de *software* livre cresce de forma afirmativa. É importante destacar a implementação do PSSL (Programa Serpro de Software Livre), instituído pelo Serviço de Processamento de Dados do Governo do Brasil, como também da página web [softwarelivre.gov](http://softwarelivre.gov). O primeiro com o objetivo de prover direcionamentos para o uso e o desenvolvimento de soluções em *software* livre no âmbito do poder público, através da atuação do, anteriormente referido, Comitê Permanente Executivo e de comitês regionais, e o segundo objetivando divulgar soluções, casos de sucesso e promover debates. No “portal” supracitado, o Governo expõe como casos de sucesso de migração para *software* livre, em níveis variados, as seguintes instituições públicas: Banco do Brasil; Serpro; Embrapa; Dataprev; Ministério das Comunicações; Exército; Radiobrás, Ministério do Desenvolvimento Agrário e Tribunal Regional do Trabalho da 4ª Região (INSTITUTO DE TECNOLOGIA E INFORMAÇÃO, 2011).

Na edição do dia 19 de janeiro de 2011 do Diário Oficial da União, o governo federal, através da Secretaria de Logística de TI do Ministério do Planejamento, publicou uma instrução normativa proibindo o uso de componentes, ferramentas, códigos fontes e utilitários proprietários, além da dependência de um único fornecedor.

Atualmente, no país, existe um número considerável de instituições e associações interessadas na divulgação e debate em torno do *software* livre, a exemplo da ASL (Associação *Software* Livre.org), uma das organizadoras do FISL (Fórum Internacional do *Software* Livre), que em junho/julho de 2011 teve a sua décima segunda edição, reunindo mais de 6,9 mil participantes de vários países no Centro de Eventos da PUCRS (Pontifícia Universidade Católica do Rio Grande do Sul) (FISL, 2011). Em 2003 o IV FISL foi marcado pela criação do PSL-BR (Projeto *Software* Livre Brasil).<sup>34</sup> Diante desse cenário, que reflete uma tendência mundial, surgem no Brasil iniciativas de entidades de educação voltadas para a promoção de linhas de ensino, pesquisa e desenvolvimento de *software* livre, a exemplo do CCSL (Centro de Competência em *Software* Livre) do Instituto de Matemática e Estatística da USP (Universidade de São Paulo), que integra a rede internacional de Centros de Competência do Projeto QualiPSO (Quality Platform for Open Source Software), a qual, além do Brasil, possui representações na Espanha, Alemanha, Itália, França, China e Japão.

---

<sup>34</sup> Para maiores informações sobre a entidade consultar o *website* <[www.softwarelivre.org](http://www.softwarelivre.org)>

## **2. EVOLUÇÃO DA COMPUTAÇÃO E PROGRAMAS GRÁFICOS**

## 2.1 O Uso do Computador na Área Gráfica

O projeto, enquanto ato de planejar a realização de um objeto ou edificação, ao longo da história, revelou-se de muitas formas, utilizando diferentes instrumentos e técnicas para representar o elemento a ser erguido, bem como mensurar e quantificar os insumos necessários à sua materialização. De acordo com Barr e Juricic (1994), a fundamentação científica deste tipo de atividade iniciou-se em 1795, com a publicação do Tratado da Geometria Descritiva, escrito por Gaspard Monge. Monge mostrou que muitos dos problemas espaciais poderiam ser resolvidos graficamente, através do uso de dois ou mais planos de projeção. Desde então, o desenho projetivo tornou-se a metodologia preferencial para a representação do projeto, conservando a utilização de instrumentos e técnicas manuais por cerca de 200 anos, até o desenvolvimento da tecnologia computacional (BARR; JURICIC, 1994). Essa tecnologia tem origem na busca humana por um meio de realizar cálculos através de dispositivos físicos ou máquinas, prática que se tornou conhecida como processamento de dados (PIRES, [200-]). Desde o surgimento da primeira máquina de calcular, até a invenção do computador<sup>35</sup>, e seu emprego efetivo na automação de processos em diversas áreas da atividade humana, foram necessários muitos avanços em termos de hardware<sup>36</sup>, desencadeados principalmente pelo desenvolvimento da eletroeletrônica e, em termos de software, pelo desenvolvimento das técnicas e linguagens de programação, de forma que somente entre as décadas de 1980 e 1990 os avanços simultâneos nessas duas áreas marca de forma mais contundente o desenvolvimento da tecnologia computacional rumo à popularização do uso do computador e sua consolidação como ferramenta de auxílio ao projeto.

Segundo Eastman (1999), os primeiros passos para o uso do computador como ferramenta de auxílio a projetos se deu em meados dos anos 1950, com desenvolvimento de aplicações isoladas na área de estruturas, quantificação, orçamento (EASTMAN, 1999), e redes de controle tipo PERT e CPM<sup>37</sup> (CARDOSO, 1986). De acordo com Lawson (1999), os passos

---

<sup>35</sup> Blaise Pascal (1623-1662), matemático, físico, escritor e artista francês, inventou a primeira máquina de calcular do mundo. Charles Babbage (1792-1871) inventou na Inglaterra o primeiro computador mecânico com o *pilot model or prototype of the difference engine*, posteriormente, "*analytical engine*", que não era simplesmente uma máquina de calcular, porque imprimia tabelas, estocava números e alterava instruções. (PIRES, [200-])

<sup>36</sup> Conjunto de unidades físicas, componentes, circuitos integrados, discos e mecanismos que compõem um computador ou seus periféricos.

<sup>37</sup> Os termos PERT e CPM são acrônimos de *Program Evaluation and Review Technique* (PERT) e *Critical Path Method* (CPM). Usadas em planejamento, as técnicas denominadas PERT e CPM foram

seguintes foram direcionados na busca por evidenciar como os computadores poderiam auxiliar na concepção de projetos, com a tentativa de utilizá-los na formulação de arranjos físicos, alternativas de pré-dimensionamento e distribuições espaciais que conduzissem a soluções que atendessem a características e restrições específicas. Entretanto, na década de 1950, apesar das investidas no uso do computador na área de projeto, a tecnologia da computação era dominada por poucos indivíduos e faltava uma formulação teórica sistematizada para esse fim (CARDOSO, 1986). Para Cardoso (1986), as pesquisas de Christopher Alexander<sup>38</sup>, iniciadas em 1962 nos EUA, sobre a sistematização de projetos arquitetônicos foi um dos trabalhos mais importantes para a construção dessa formulação.

Muitos autores, a exemplo de Besant (1985), Cardoso (1986) e Eastman (1999), são unânimes em afirmar que o primeiro passo importante para a introdução efetiva do uso dos computadores na área de projetos se deu em 1963 no MIT<sup>39</sup> (*Massachusetts Institute of Technology*), quando foi demonstrado o Sketchpad (figura 2), ou sistema de gráficos interativos, desenvolvido por Ivan E. Sutherland, e assim chamado por se tratar de um sistema que trabalhava através de *inputs*, ou entradas, realizadas pelo usuário através de uma “caneta de luz” permitindo a criação e manipulação de desenhos que apareciam como informações gráficas exibidas em um monitor, recém desenvolvido, que funcionava com base em um osciloscópio de raios catódicos<sup>40</sup>, acionado por um computador. A demonstração do Sketchpad requereu a utilização de um dos *mainframes*<sup>41</sup> do MIT (EASTMAN, 1999). Esse sistema, de utilização ainda restrita, levantou a possibilidade de um uso mais efetivo da computação na área de projeto, justamente onde as habilidades humanas são inferiores às do computador: em velocidade, precisão e armazenamento de dados. Surge, então, o que se conhece hoje como CAD (*Computer Aided Design*) ou Projeto Auxiliado por Computador.

---

independentemente desenvolvidas para o planejamento e controle de Projetos nos anos 1950, porém a grande semelhança entre estas fez com que o termo PERT/CPM fosse utilizado corriqueiramente como apenas uma técnica. Disponível em: <<http://pt.wikipedia.org/wiki/PERT>> Acesso em: 11 dez. 2006.

<sup>38</sup> Christopher Alexander, nascido em Viena na Áustria em 1936, mestre em matemática e bacharel em arquitetura pela Universidade de Cambridge – EUA, e PHD em arquitetura pela Universidade de Harvard. Disponível em: <<http://www.patternlanguage.com/leveltwo/ca.htm>>. Acesso em: 21 jun. 2008.

<sup>39</sup> O *Massachusetts Institute of Technology*, mais conhecido pela sua sigla MIT, é um centro universitário de educação e pesquisa localizado em Cambridge, nos EUA. O MIT é um dos líderes mundiais em ciência e tecnologia, bem como outros campos, como administração, economia, lingüística, ciências políticas e filosofia.

<sup>40</sup> O osciloscópio de raios catódicos consiste em um Tubo de Raios Catódicos (CRT – *Catodic Ray Tube*), que produz um feixe de elétrons que atinge a camada de fósforo da parte detrás da tela do monitor, fazendo com que esse ponto brilhe (BESSANT, 1985).

<sup>41</sup> *Mainframe* é o nome dado para computadores de grande porte, dedicados normalmente ao processamento de um volume grande de informações. Geralmente ocupam um grande espaço e necessitam de um ambiente especial para seu funcionamento. Disponível em: <<http://pt.wikipedia.org/wiki/Mainframe>>. Acesso em: 22 jun. 2008.

Segundo Cardoso (1986), essa designação foi dada em 1963 por S. A. Coons, pesquisador do MIT, quando percebeu que o computador podia, a partir das novas possibilidades gráficas, ser aplicado na resolução de problemas de projeto, principalmente no que dizia respeito à representação. Dessa forma, foi mesmo a representação da geometria o motivo principal que levou ao uso do computadores em projeto e fabricação de produtos (EASTMAN, 1999).



**Figura 2** – Sketchpad em uso.  
Fonte: Sutherland (2003, p. 20).

Os principais recursos que o Sketchpad permitia eram:

- Uso da “caneta de luz” como dispositivo de entrada de dados;
- Uso do monitor de vídeo como dispositivo de saída de dados;
- Uso de instruções específicas pré-programadas para o traçado de elementos geométricos básicos, como retas, e arcos de circunferências;
- Montagem de arquivos com elementos gráficos, que podiam ser resgatados ou alterados (BESANT, 1985).

Cardoso (1986), destaca os seguintes eventos, ocorridos nessa época, que mais tarde exerceriam influência no surgimento de sistemas CAD direcionados especificamente para o campo da arquitetura:

- 1 - Realização de um curso em Oxford sobre computação especificamente para arquitetos

(1963);

- 2 - Estudos sistematizados sobre o problema do arranjo físico em projetos de arquitetura iniciado pelo trabalho de Whitehead & Eldras, na Inglaterra (1964);
- 3 - Realização da Conferência sobre Computação e Arquitetura – *Architecture and the Computer – First Architectural Center Conference* – Boston – 1964;

A partir do desenvolvimento do Sketchpad, e com o desenvolvimento dos equipamentos periféricos e linguagens de programação, surgiram os primeiros sistemas gráficos comerciais:

24. 1964 – Desenvolvido pela IBM e General Motors o primeiro sistema gráfico de uso comercial, o DAC-1 (*Design Automated by Computer*).
25. 1965 – Surge o segundo sistema gráfico comercial – GRAPHICA 1, desenvolvido pelos laboratórios da indústria norte americana Bell, já com um conceito de estação de trabalho ligado a um computador central, *on line*.

Esses sistemas eram caros por serem sofisticados e exigirem muito dos recursos e potência do computador. Foram, inicialmente, desenvolvidos para aplicações em áreas ligadas ao projeto de componentes eletrônicos, mecânicos, hidráulicos, navais, automobilístico, aeroespaciais, militares e aeronáuticos, para obter maior precisão na representação de componentes complexos e aumentar a velocidade nas linhas de produção, diminuindo os custos e trazendo mais valia ao processo produtivo industrial. Portanto, eles foram adotados apenas pelas principais indústrias, onde a sua utilização nos projetos justificava o alto custo do investimento (BESANT, 1985)

No decorrer da década de 1960, congressos, publicações e conferências faziam fervilhar a idéia da “automação do projeto” no meio profissional e acadêmico da engenharia e da arquitetura (CARDOSO, 1986). A possibilidade de utilizar a tecnologia computacional a serviço da elaboração de projetos de edifícios dava ao processo projetual uma nova perspectiva, ainda que efetivamente mais restrita ao campo da representação e alguns tipos de análises. Nesse período, começaram a surgir várias aplicações em áreas técnicas específicas relacionadas com o projeto do edifício, tais como: estrutura, acústica, iluminação, conforto térmico, circulação vertical, orçamento e legislação das construções. Com os novos avanços alcançados no campo do *hardware*, entre eles os dispositivos de entrada e saída de dados gráficos, acompanhado pelo avanço constante na formulação de novas linguagens de

programação, e padrões de arquivo intercambiáveis, uniu-se o campo do CAAD<sup>42</sup> (*Computer Aided Architectural Design*) de sistemas, formados por um conjunto de ferramentas aplicáveis a diversas áreas técnicas do projeto do edifício agregadas em um único programa. Esses programas eram extremamente caros e requeriam pessoal altamente especializado para operá-los, aplicavam-se perfeitamente a projetos de edifícios onde o grau de subjetividade era reduzido e a produção feita em série. Assim, inicialmente eles eram utilizados principalmente por agências governamentais dos EUA e Europa, ligadas à área da produção da habitação em massa, projetos de edificações com programas específicos a exemplo de escolas e hospitais, muitas vezes aplicados na formulação de arranjos físicos espaciais em edificações compostas de elementos pré-moldados, padronizados e produzidos em série. Cardoso (1986) relaciona os seguintes sistemas como sendo os principais deste período:

- ICES – “*Integrated Civil Engineering System*” - EUA – 1967;
- WSCC – “*West Sussex Country Council*” – Inglaterra – 1968;
- HARNESS – Inglaterra – 1969;
- ARK-2 – “*Architectural Kinetics – Man and Machine*”- EUA – 1970;
- SSHA – “*Scottish Special Housing Association*” – Escócia – 1971;
- OXSYS – “*Oxford System*” I – Inglaterra – 1972;
- CEDAR – “*Computer – aided Environmental Design Analysis and Realization*” – Inglaterra – 1972;
- SPACES 1 a 3 – Escócia – 1972.

O já mencionado desenvolvimento de novas linguagens de programação, a queda constante das limitações de memória e processamento das máquinas, fez da primeira metade da década de 1970 um período de muito progresso no desenvolvimento de programas CAD. Na segunda metade dessa década, uma verdadeira revolução na indústria dos computadores viria a desencadear um processo evolucionário onde diferentes estruturas para representação de dados geométricos surgiam a partir de novas propostas lançadas nesse emergente mercado, num esforço constante para obter vantagens sobre a concorrência, de forma que, se a inovação

---

<sup>42</sup> Projeto Arquitetônico Assistido por Computador (*do inglês CAAD, Computer Aided Architectural Design*) ou ainda, CAD para arquitetura, consiste na utilização de ambientes computacionais para resolução de problemas de projeto e representação gráfica da forma e do espaço arquitetônico.



de uma obtinha sucesso, logo era copiada pelas outras. Segundo Eastman (1999), na década de 1970, novas companhias de CAD eram formadas mensalmente: Applicon, Calma, Autotrol, Intergraph, CADAM e outras, marcado pelos seguintes avanços no campo do *software*:

- Desenvolvimento de algoritmos mais eficientes para o desenho de entidades gráficas, arrastamento e escalonamento de imagens;
- Introdução de símbolos;
- Aumento do número de *layers* (camadas de desenho);
- Linguagens de aplicações;
- Hachuras e cotagem associativas;
- Modelamento sólido;
- Modelamento sólido paramétrico;
- Módulos integrados de renderização<sup>43</sup>.

Todos os fatores demonstrados por Cardoso (1986) e Eastman (1999), e aqui relatados, acabaram por promover o aperfeiçoamento dos programas CAD, que passaram a oferecer maior flexibilidade na forma de acionar a sequência de comandos na construção de gráficos, operações de transformação e maior auxílio na visualização e composição de formas tridimensionais, além disso, os profissionais de projeto passaram gradativamente a participar do processo de elaboração destes programas, tornando-os mais práticos. Entretanto, foi o desenvolvimento e os avanços alcançados pela indústria do PC<sup>44</sup> que possibilitou o amplo acesso a esses sistemas, que a partir da década de 1980 iria reverter a situação de frustração da expectativa da popularização do uso do computador como ferramenta efetiva de auxílio ao projeto do edifício.

## 2.2. A Popularização do Uso do Computador como Ferramenta de Auxílio ao Projeto

O lançamento promissor, em 1981, do IBM *Personal Computer*<sup>45</sup> (figura 3) e o

<sup>43</sup> O termo “renderizar” provém do inglês *to rende*, que em computação gráfica é o nome dado ao processo pelo qual se pode gerar imagens digitais, ou *raster image*, a partir de um modelo geométrico tridimensional, simulando condições reais de cor, materiais, textura e iluminação.

<sup>44</sup> Trata-se de um computador de dimensões reduzidas, que se destina ao uso pessoal (ou para uso de um pequeno grupo de indivíduos). A expressão “computador pessoal” é geralmente abreviada para a sigla PC, cujo significado em inglês é *personal computer*.

<sup>45</sup> O Apple II, lançado em 1978, foi o primeiro PC popular e alguns sistemas gráficos precários foram desenvolvidos para esta máquina.

estabelecimento do PC-DOS<sup>46</sup> como Sistema Operacional dessas máquinas atraiu a atenção de empresas produtoras de *softwares* utilitários e em pouco tempo a maior parte desses programas estavam sendo produzidos para “rodar” sobre esta plataforma. Com uma “arquitetura aberta”, ou seja, seus componentes podiam ser substituídos por similares produzidos por outras companhias, o IBM PC ajudou também a acionar o desenvolvimento de outras companhias que passaram a oferecer uma gama variada de recursos de extensão para vídeo, ou memória adicional.



**Figura 3** – IBM PC, versão original.  
Fonte: (IBM\_PC\_5150.JPG, 2011).

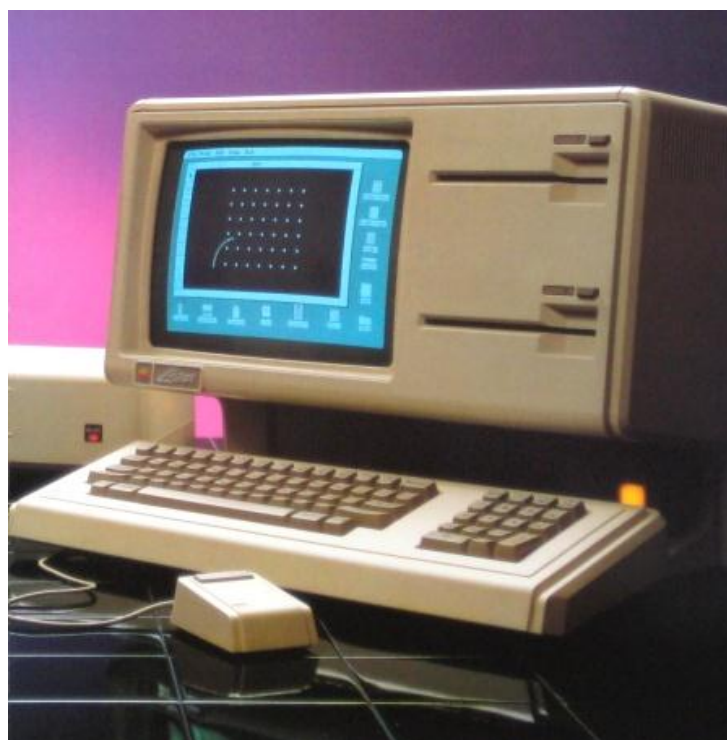
Em pouco tempo a compatibilidade com essa máquina passou a ser quase que uma regra de mercado. Nesse período, década de 1980, começaram a ser desenvolvidos simples programas CAD para os computadores pessoais. Inicialmente eles eram de baixa capacidade, utilizados apenas como editor de desenho para gráficos bidimensionais, ainda assim com muitas limitações: pouca capacidade de armazenamento e baixa velocidade de processamento, incapazes de incluir ferramentas mínimas para ser considerado um CAD efetivo. Eastman (1999), aponta as seguintes, entre as companhias pioneiras no desenvolvimento de *software* CAD para microcomputadores:

---

<sup>46</sup> O Sistema PC-DOS foi elaborado a partir de um incremento que a Microsoft fez em um sistema operacional 8086 que ela adquiriu da *Seattle Computer Products*. Este sistema atendia por diversos nomes, inclusive “Quirk and Dirt DOS”, ou QS-DOS.

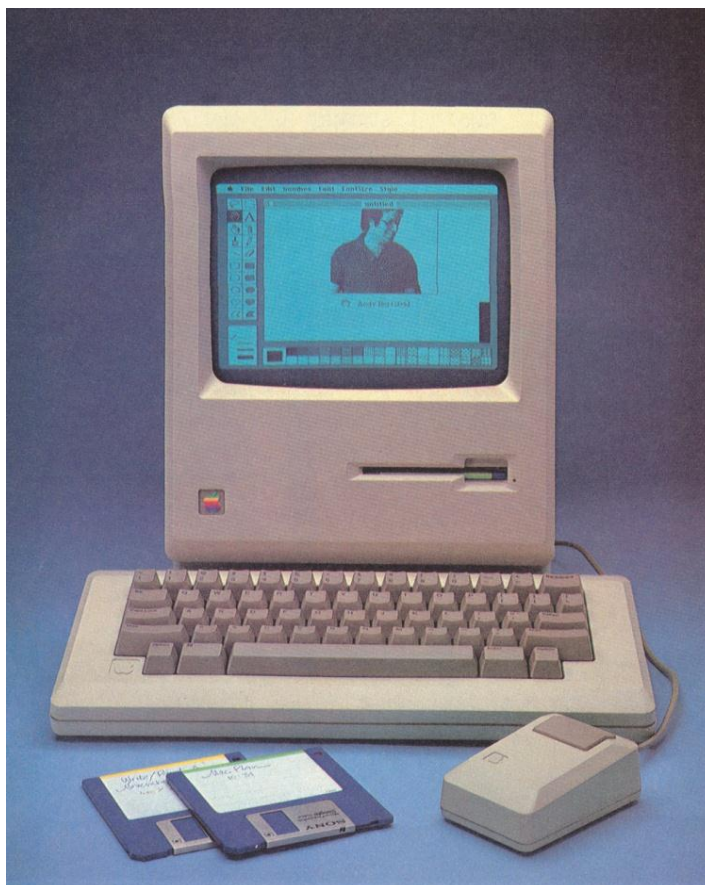
- Autodesk;
- VersaCAD;
- Summagraphics,
- Microstation.

Observa-se que as limitações apresentadas decorriam muito mais pela deficiência na capacidade de processamento, de dados e de imagem, da máquina do que pela falta de recursos para elaboração de programas mais sofisticados. Para Eastman (1999), o IBM-PC era inadequado para o uso profissional do CAD. Apesar da supremacia dessa máquina no início da década de 1980, os primeiros avanços significativos foram realizados pela Apple, que com base na tecnologia inspirada pelo trabalho do Centro de Pesquisa de Palo Alto da Xerox, insinua os grandes progressos em computação gráfica. Com a introdução do Lisa, lançado em 1983, (figura 4), surgiram as primeiras tentativas de combinar, em um produto comercial, janelas, ícones e interface gráfica com o usuário, mas em consequência do preço elevado - quase US\$ 10.000,00 - e pelo fato de a maior parte dos principais desenvolvedores de *software* estarem produzindo seus programas para o IBM-PC, não conseguiu atingir a popularização no mercado (ROMEIRO, 2004).



**Figura 4** – Apple Lisa.  
Fonte: (LISA 1- BROSHURE.JPG, [200-]).

Em 1984 a Apple lança o Macintosh (figura 5), que trazia um sistema operacional diferenciado, capaz de oferecer mais que um *prompt*<sup>47</sup> de DOS<sup>48</sup> e uma interface baseada em caracteres; ele podia abrir várias janelas para interação com programas e possuía *menus* suspensos, entretanto não atraiu tanto a atenção dos usuários de computadores pessoais, porquê não era compatível com o padrão IBM, dominante no mercado. No final de 1985 a Microsoft lança o Windows. A interface gráfica desse sistema operacional tornaria mais fácil a relação entre homem e máquina, viabilizando a aplicação massificada do uso de computadores nos mais diversos setores, atuando inclusive como elemento facilitador na interação de usuários com os programas CAD.



**Figura 5** – Apple Macintosh.  
Fonte: (APPLE MACINTOSH.JPG, 2004).

---

<sup>47</sup> Em sistemas operacionais baseados em comando, o *prompt* é constituído por um ou mais símbolos que indicam o local a partir do qual o usuário deve digitar uma instrução. No MS-DOS, por exemplo, o formato mais comum é uma letra que representa a unidade de disco, seguida do sinal “maior que”, por exemplo: C >.

<sup>48</sup> Sigla em inglês para *Disc Operating System*. Designação genérica para os sistemas operacionais “carregados” a partir de uma unidade de disco, quando um computador é inicializado. Desprovido de interface gráfica, atende a comandos digitados pelo usuário.

Em 1985, a Intel lança o microprocessador 80386 de 16 MHz<sup>49</sup>. Este novo componente trouxe novas possibilidades em poder de processamento viabilizando, em parte, a migração de muitas das capacidades dos programas CAD, que, até então, só eram disponíveis em *workstations*, para os computadores pessoais. Contudo o poder de processamento não foi a única demanda que as ferramentas CAD colocaram nos microcomputadores. Apesar de todos os avanços alcançados na área da computação gráfica, em termos de interfaces com o usuário, o uso efetivo do CAD em microcomputadores só se tornou prático apenas após o processador de imagem e monitores alcançarem a resolução SVGA<sup>50</sup> (800x600) *pixels*, e capacidade de exibição simultânea de 256 cores (EASTMAN, 1999).

Segundo Eastman (1999), em meados da década de 1980, grupos de pesquisa em várias universidades debruçaram-se sobre o problema de tornar a representação de formas tridimensionais mais simples e permitir um nível mais alto de operações para editá-las, dando origem a operações específicas e novos métodos de construção de modelagem geométrica, a exemplo das operações *booleanas*<sup>51</sup>, que acabaram por permitir uma completa representação tridimensional, e possibilitar um alto nível de ações para descrição da forma do objeto, além de servir de entrada para vários tipos de análises. Desenhos passaram a ser automaticamente gerados como informação através de secções ou projeções de um modelo geométrico tridimensional, e o fato de serem extraídos de um único modelo garantia a coerência entre eles.

Com todas essas inovações nos campos do *hardware* e do *software*, operações de modelagem de sólidos e superfícies, várias operações de edição que requeriam visualização dinâmica das entidades gráficas, e capacidades de “renderização” de imagens passaram a fazer parte dos

---

<sup>49</sup> O hertz (símbolo Hz) é a unidade derivada do Sistema Internacional de Unidades para medir frequência. Megahertz (símbolo MHz) é a medida para a velocidade de processamento de um computador. Um sinal de 1 MHz significa que ele possui 1 milhão de ciclos em um segundo. A duração do ciclo é chamada período e é o inverso de sua frequência. Em um sinal de 1 MHz, cada ciclo dura 1 microssegundo (1 microssegundo é igual a 0,000001 segundo). Disponível em: <<http://www.clubedohardware.com.br/dicionario/termo/162>>. Acesso em: 23 jun. 2008.

<sup>50</sup> Super Video Graphics Array - Este nome é usado para qualquer placa de vídeo que utilize padrões de vídeo acima do VGA e que tenha total compatibilidade com o padrão VGA. É o padrão encontrado em todas as placas de vídeo atuais, pois é capaz de representar várias resoluções, sendo as mais comuns as de 800x600, 1024x768 e 1280x1224. Disponível em: <<http://www.clubedohardware.com.br/dicionario/termo/279>>. Acesso em: 23 jun. 2008.

<sup>51</sup> CSG – Constructive *Solid Geometry*, técnica utilizada para modelagem de formas tridimensionais, com operadores, que permite ao modelador criar formas complexas a partir de operações booleanas. Dadas duas formas gera a união, interseção ou diferença delas. Pelo resultado ser sempre uma forma fechada, permite que a operação seja repetida. Disponível em: <[http://en.wikipedia.org/wiki/Constructive\\_solid\\_geometry](http://en.wikipedia.org/wiki/Constructive_solid_geometry)>. Acesso em: 23 jun. 2008.

novos pacotes CAD para microcomputadores. Com a introdução dessas novas possibilidades o “CAD *desktop*”<sup>52</sup> se tornou uma realidade, seu potencial mercadológico se expandiu consideravelmente tornando-os viáveis para o uso em escritórios de arquitetura e engenharia de médio e pequeno porte e, até mesmo, para o uso individual. O aperfeiçoamento das interfaces gráficas, cada vez mais intuitivas, contribuiu em muito para descomplicar a utilização dos programas, podendo ser facilmente operados por engenheiros e arquitetos. A proliferação destes programas, liderado em muitos setores da indústria da construção pelos produtos da *Autodesk*, e seguido por muitos outros desenvolvedores, estimulou a aparição de diversos centros de treinamento e literatura farta sobre o assunto. Assim, o final dos anos 80 e início dos anos 90 foi o período em que o mercado de CAD se consolidou e expandiu. Hoje as aplicações CAD estão disponíveis para qualquer um que tenha acesso a um microcomputador.

Como forma de resumir e sintetizar a evolução histórica do uso do computador na área gráfica, descrita anteriormente, consta do APÊNDICE A gráficos, na forma de linha do tempo, com os principais eventos na evolução das tecnologias de *hardware* e *software*, assim como os fatos marcantes, entre as décadas de 1950 e 1990, rumo à consolidação do uso do computador no campo do projeto.

### 2.3 Classificação das Ferramentas CAD

Romeiro (1997), atribui à Tecnologia CAD um caráter multidisciplinar pelo fato de ser possível aplicá-la em diversas áreas onde existe uma interação do computador com a atividade de projeto. Besant (1985), aborda aspectos importantes da relação existente entre usuário e computador, destacando a integração desse dois elementos em um processo interativo no qual o homem atua onde a máquina é menos eficiente e vice-versa, visando a resolução de problemas agrupando as melhores características de cada um.

Analisando as referências feitas a esses dois autores, pode-se definir um sistema CAD, de forma simplificada, como um sistema computadorizado composto de *hardware*, *software* e usuários capazes de operá-lo adequadamente para fins da elaboração de projetos. Nunes (1997) chama atenção para a quantidade e qualificação dos recursos humanos que devem

---

<sup>52</sup> *Desktop* são os computadores de mesa, estes que tem tudo separado (*mouse*, teclado, CPU ou gabinete, monitor, impressora...) e são os de menor custo. Disponível em: <<http://pt.wikipedia.org/wiki/Desktop>> Acesso em: 23 jun. 2008.

integrar um sistema CAD, admitindo ser fundamental contar com profissionais que dominem as disciplinas e especialidades envolvidas na projeção, os quais se encarregarão das atividades de produção e definições requeridas no processo projetual, e com usuários dotados de conhecimento para operar corretamente o sistema, no que se refere à manipulação dos dispositivos de *hardware* e dos programas utilizados. Dependendo do porte do projeto e do aparato tecnológico escolhido, pode também envolver analistas de suporte e operadores de recursos de informática. A estratégia de adoção e implantação do sistema, envolve questões de produtividade (custo/benefício), que são tratadas a nível gerencial. *Este nível deve ter sob sua responsabilidade ou coordenação a atualização, o aperfeiçoamento e o treinamento dos recursos humanos* (NUNES, 1997, p. 24). Nunes (1997) apresenta a tabela com os componentes básicos de um sistema CAD:

**Quadro 3 – Composição do sistema CAD**

COMPOSIÇÃO DE UM SISTEMA CAD	
Recursos Humanos	Gerência Engenharia (projeto e suporte) Operacional (projetista/desenhista)
Recursos de Informática	<i>Hardware</i> (equipamentos)
	<i>Software</i> (programas)

Fonte: Nunes (1997, p. 24).

Analisando o quadro 3, observa-se que o *software* é uma ferramenta que compreende parte do sistema computacional de auxílio ao projeto. Num sentido lato *software* é uma expressão que se utiliza em contraste com *hardware*, para referenciar todos os programas e procedimentos relacionados que podem ser utilizados em um sistema de computador, para sua exploração, requerendo equipamentos com poder computacional capaz de processar e armazenar os dados em sintonia com as necessidades requeridas pelo programa.

Nota-se que é comum para os leigos associarem programas CAD com editores de desenho. O *Computer Aided Design*, além do editor geométrico, reúne um conjunto de ferramentas para a organização, gestão e produção de projetos, podendo interagir com outros programas que realizam análises diversas, geram informações e podem auxiliar na fabricação de componentes a partir de modelos matemáticos gerados com a ferramenta CAD. Segundo Amorim (1997), diversas ferramentas de auxílio ao projeto foram desenvolvidas com base na Tecnologia CAD,

agregando-se recursos de outras tecnologias, dando origem a sistemas integrados, multifinalitários, aplicáveis a diferentes campos da atividade projetual. Entre essas tecnologias Amorim (1997) destaca as seguintes:

- omputação Gráfica;
- ecnologia de Geoprocessamento
- E – Sistemas Especialistas;
- V – Realidade Virtual, e
- IC – Tecnologia da Informação e Comunicação.

Considerando as possibilidades de emprego dessas tecnologias, bem como a vasta gama de ferramentas CAD existentes, uma forma usual de classificação desses programas, que se presta adequadamente aos propósitos deste estudo, é pela natureza de sua aplicação, agrupando-os em duas categorias: ferramentas genéricas e ferramentas dedicadas.

## **2.4 Ferramentas CAD Genéricas**

Pode-se afirmar que ferramentas CAD genéricas são aquelas indistintamente aplicáveis em áreas diversas no auxílio à produção e gerenciamento de projetos, a exemplo da indústria de manufaturas, automobilística, naval, aero-espacial, eletromecânica, engenharia e arquitetura, ou seja, não estão orientados para uma determinada aplicação, Amorim (1999). Esses *softwares* podem variar de simples editores de desenho, até sofisticados modeladores geométricos tridimensionais parametrizados, dotados de recursos para criação, visualização, documentação, compartilhamento de informações e uma linguagem de programação para a construção de aplicações específicas relacionadas com o ramo da atividade projetual no qual serão empregados. Para Besant (1985), esse último recurso surgiu da necessidade de se cobrir a vasta gama de aplicações, dada a natureza multidisciplinar e a não padronização dos elementos do projeto. Cordeiro (1996) ressalta que os programas CAD genéricos são



normalmente utilizados para subsidiar processos que, feitos de forma não automatizada, seriam lentos e intensivos de mão de obra, para redução de custos e aceleração das operações pertinentes do processo projetual. O representante mais conhecido dessa categoria de programa CAD é o AutoCAD, da Autodesk.

## 2.5 Ferramentas CAD Dedicadas

Em contraponto aos programas genéricos, as ferramentas CAD dedicadas são aquelas orientadas para automatização de tarefas específicas, incorporando recursos disponíveis em programas CAD de uso genérico, mas, principalmente, são ferramentas orientadas a uma determinada área de projeto, como arquitetura, engenharia mecânica, engenharia estrutural, projeto de vias etc. (AMORIM, 1999). Estes são programas que objetivam automatizar um número maior de tarefas, dispondo de recursos que atendem a um escopo mais restrito na área específica de aplicação. No caso de aplicações orientadas para arquitetura, um dos representantes mais conhecidos é o AutoCAD Architecture, da Autodesk. Desenvolvido para arquitetos e construído sobre o AutoCAD, disponibiliza funcionalidades que visam o aumento da produtividade e melhor coordenação dos desenhos, dispondo de recursos que articulam o modelo geométrico tridimensional com a documentação do projeto (AUTODESK, c2010). Alterações feitas em qualquer uma destas instâncias são automaticamente absorvidas pelas demais. Além de trabalhar com entidades básicas da geometria, trabalha principalmente com objetos arquitetônicos, a exemplo de portas, janelas, paredes, escadas e coberturas, que comportam-se segundo propriedades específicas. Amorim (1999), acrescenta a existência de módulos de extensão para programas genéricos que lhes atribuem funcionalidades de programas dedicados, chamados popularmente, segundo o autor de forma inadequada, de aplicativos.

Assim, dada a vasta gama de aplicações da tecnologia CAD, se faz necessário estabelecer um recorte que limite o universo de investigação do presente estudo. De acordo com Cardoso (1986), o Projeto do Edifício Assistido por Computador (PEAC), como ficou conhecido, utiliza principalmente sistemas CAD voltados especificamente para a elaboração de projetos de arquitetura, sendo internacionalmente denominado como *Computer Aided Build Design* (CABD), ou ainda *Computer Aided Architectural Design* (CAAD).

Dessa forma, o estudo ora em andamento busca identificar ferramentas gráficas livres, a exemplo dos programas CAD de natureza genérica, ou dedicada, que se enquadrem nessa categoria de *software*, e cujos recursos possibilitem a sua aplicação, de forma parcial ou integral, ao atendimento das demandas do PEAC, em suas diversas fases de progressão.

Considerando o viés de desenvolvimento do trabalho em curso, torna-se necessário tecer algumas considerações sobre a natureza do processo de projeto e suas perspectivas enquanto atividade coletiva e multidisciplinar, uma vez que o elenco e a escolha das ferramentas computacionais a serem utilizadas no seu desenvolvimento estão atreladas a esses aspectos da projeção.

## **2.6 Considerações Sobre a Sistemática do Projeto de Arquitetura**

Ao discutir a natureza do processo de projeto no artigo intitulado: *Linguagem, Informação e Representação do Espaço*, Amorim (2000) levanta o debate frequente sobre o caráter racional e científico do procedimento projetual, concluindo que trata-se de um processo sujeito à aplicação de normas e de uma sistematização que objetiva o controle das diversas fases de seu desenvolvimento, fases essas que compreendem coleta e análise de dados, síntese de idéias, verificação de resultados e descrição da proposta.

Dentro desse mesmo viés, Naveiro e Borges (1998) buscam, na área da Sociologia da Inovação, uma abordagem que trata o projeto como um processo coletivo, viabilizado pela organização que o sustenta, cujos principais problemas envolvidos são: [...] *criação de um ambiente para o trabalho intelectual cooperativo, gerenciamento e controle do estado da informação* [...] (NAVEIRO; BORGES, 1998, p.52). Dentro dessa perspectiva, os autores consideram quatro aspectos básicos presentes no processo de projeção:

- Organização – Envolve principalmente a identificação das tarefas elementares, da relação que se estabelece entre os participantes, e da interface clara entre as diferentes disciplinas envolvidas no processo;
- Universos de Competência – Refere-se ao caráter multidisciplinar do projeto, isto é, os diferentes ambientes de especialização técnica, com seus "dialetos" e símbolos próprios

- Condicionantes – São as especificações e restrições que delimitam os diversos constituintes do projeto. A definição de como essas restrições serão atendidas é parte integrante do projeto;
- Discurso – Refere-se à linguagem utilizada, modelos e metáforas para representação de soluções “abstratas” e a criação de interfaces com os demais contextos que influenciam o universo de competência do projetista.

Naveiro e Borges (1998) definem, de forma generalizada, o projeto como uma atividade de resolução de problemas que envolve diversos atores, diferentes especialidades, contextos e condicionantes, necessitando de uma sistemática organizacional e de uma linguagem apropriada que promova a interação entre os membros da equipe e conduza, de forma progressiva, ao produto final. Essa formulação corrobora com a referência feita a Amorim (2000), utilizada anteriormente. Para Naveiro e Borges (1998), a estruturação do entendimento das categorias de informação contidas no universo do projeto é determinante para o ordenamento do processo e para a correta escolha e utilização dos meios informatizados que auxiliarão no seu desenvolvimento. Por outro lado, observa-se em Cardoso (1986) e em Kalisperis (1996) que os estudos e debates que levaram ao entendimento e à sistematização do processo de projeto no campo da arquitetura evoluíram, em grande parte, decorrente das contribuições feitas por estudos outros, desenvolvidos nas décadas de 1960 e 1970, que buscavam uma forma de introduzir o uso do computador na atividade projetual, na perspectiva de automação do mesmo, em busca, inclusive, de utilizá-los como “máquinas pensantes”, capazes de apresentar soluções de projeto, inspirados pelas experiências pioneiras, bem sucedidas, de aplicação dos computadores na engenharia.

No contexto das considerações feitas anteriormente, Romeiro (1997) traz uma abordagem sobre o papel integrador dos sistemas CAD no processo de projeto, que embora seja genérico, ou seja, não trata especificamente de uma determinada área de aplicação do CAD, serve perfeitamente para traçar um paralelo de como o auxílio computacional é absorvido ao longo da progressão do projeto de arquitetura. O autor esquematiza essa absorção em cinco fases distintas, a seguir descritas e interpretadas:

- Primeiramente utiliza-se programas voltados para a representação do projeto, *software* CAD, e programas para realização de processos isolados de cálculo, esses últimos, lembra o autor, pertencentes à classe de programas de auxílio à engenharia, ou CAE

(*Computer Aided Engineering*). Fazendo um paralelo com o projeto de arquitetura, seria equivalente aos primeiros desenhos e cálculos, que consideram partes isoladas da edificação, como as primeiras representações da geometria do edifício e ensaios estruturais, ou do sistema de abastecimento de água e esgotamento sanitário.

- Num segundo momento, o autor diz dar-se início à realização de procedimentos interativos controlados pelo projetista. Refere-se ao processo de tomada de decisões no qual o homem atua onde a máquina é menos eficiente e vice-versa. Nesse estágio, ocorre a definição das alternativas que atenderão melhor aos requisitos específicos, baseado nos resultados obtidos na fase anterior, envolvendo aspectos quantitativos e qualitativos, esses últimos mais relacionados com a percepção e o julgamento humano.
- Na fase seguinte, Romeiro (1997) diz haver sistemas integrados para todo o processo de desenvolvimento de projetos, baseados em modelos numéricos mais completos e sofisticados. Esse estágio equivale à compatibilização do projeto de arquitetura com os demais projetos complementares e a realização de simulações que necessitam de informações que considerem a integração dos sub-sistemas que compõem a edificação, a essa altura reunidos em um único modelo, a exemplo de simulações acerca da condutividade térmica, consumo de energia e eventuais impactos ambientais, levando-se em conta o meio onde o edifício será inserido.
- A seguir, considera-se a integração entre sistemas para desenvolvimento de projetos e sistemas análogos que utilizam as informações do modelo construído anteriormente, mas dirigidos para problemas específicos de produção e controle. Equivale à produção da documentação do projeto executivo e utilização de ferramentas para quantificação de materiais, insumos, custos e estratégias de execução da obra. Observa-se, entretanto, que a integração entre projeto e produção no campo do AEC<sup>53</sup> ainda não alcançou o nível de integração atingido por outros ramos da Indústria, a exemplo da indústria aero-espacial e automobilística.
- Por fim, a integração do sistema informacional técnico assim obtido, que unifica e

---

<sup>53</sup> Do inglês, *Architecture, Engineering and Construction* – Arquitetura Engenharia e Construção.

reúne em um só sistema as várias fases de desenvolvimento de projetos, de produção e de controle, com um sistema informacional, gerencial e administrativo para formar um único sistema empresarial integrado. Representa um estágio mais avançado de automação. Essa também é uma realidade que se configura de forma mais efetiva no campo da manufatura, viabilizada pela padronização de componentes e produção em série, mas tem sido um campo de discussão no âmbito do desenvolvimento de novas tecnologias computacionais aplicadas ao projeto de arquitetura.

Diante do exposto, conclui-se que as aplicações computacionais para a área do projeto de arquitetura são empregadas na perspectiva da criação de um ambiente de trabalho munido de ferramentas capazes de auxiliar na construção de um modelo que carregue, tanto quanto possível, as informações sobre as características físicas do elemento a ser erguido e determinadas propriedades das partes que o compõe, atendendo às especificidades das disciplinas nele envolvidas e utilizando uma “linguagem” que promova a interface entre elas, permitindo, assim, avaliar sua adequação aos diversos contextos aos quais será submetido, propiciando a tomada de decisões, organização do processo projetual, planejamento e controle da produção.

## **2.7 Principais Aplicações Para a Área do Projeto de Arquitetura**

Para avaliar a aplicação dos *softwares* livres gráficos na arquitetura se faz necessário contextualizar a forma como o auxílio computacional tem sido empregado na prática projetual. Ratford (1987, apud KALISPERIS, 1996) considera as seguintes aplicações dos computadores na área do projeto de arquitetura: representação, simulação, geração e otimização. Apesar de ter sido formulada há mais de vinte anos, essa classificação será utilizada como referência, uma vez que não contradiz as práticas atuais, apesar da evolução da tecnologia e das ferramentas CAD, mas também por serem compatíveis com o atendimento das necessidades identificadas anteriormente ao se discutir a sistemática do projeto de arquitetura. Nesse contexto, as referidas aplicações serão detalhadas a seguir.

### **Aplicações de Representação:**

Kalisperis (1996), diz que a representação em arquitetura é um meio não apenas para comunicar idéias, mas também para auxiliar no próprio processo de definição das soluções a serem adotadas. Dessa forma, a atividade projetual requer um conjunto de representações e,

não obstante, a representação foi e continua sendo uma das áreas centrais de estudo e desenvolvimento das ferramentas CAD. Mitchel (2008) ao discutir as maneiras de materialização da forma, pré-existente na mente do projetista, através das “mídias” de representação, introduz o conceito de mundos projetuais e de modelo na arquitetura: [...] *um modelo é uma coletânea de símbolos gráficos, pontos, linhas e polígonos, que formam um arranjo bi ou tridimensional* (MITCHELL, 2008, p.52). Ao fazer essa afirmativa, o autor diz que se pode pensar em um espaço povoado por esses símbolos como sendo um mundo projetual. Nesse mundo, esses símbolos podem ser manipulados para esboçar possibilidades para o mundo real. Isso significa dizer que cada mundo projetual disporá de instrumentos e técnicas apropriadas para criação e manipulação do modelo. Assim, ao se optar pela utilização de um sistema CAD o projetista lida com um mundo projetual formado por uma estrutura de dados gráficos com operações a ela associadas que permitirá a manipulação dessa estrutura e a tradução da mesma em uma imagem projetada no monitor, correspondente à geometria do modelo, seja ela tridimensional ou bidimensional, ou ambas, conforme as características do sistema.

Aos símbolos gráficos; (pontos, linhas e polígonos), citados na definição de modelo introduzida por Mitchell, se atribui a denominação de unidades primitivas, ou símbolos gráficos básicos que povoam o mundo projetual, a partir dos quais, através de procedimentos diversos, são feitas composições para definição, além de desenhos, de superfícies (planos) e volumes (sólidos), que, devidamente combinados, irão descrever a forma do edifício. Em um sistema CAD, a estrutura de dados é armazenada em um banco de dados relacional que, por meio de apontadores ou tabelas, permite estabelecer relações entre formas, dando ao projetista a capacidade de realizar observações sobre um projeto (MITCHELL, 2008). Para o autor, o uso prático de um mundo projetual depende do leque de observações que ele nos permite fazer a respeito de um dado projeto.

No que tange especificamente à representação, um sistema CAD configura um mundo projetual, dotado de ferramentas que podem variar de simples editores de desenho (2D) até sofisticados modeladores tridimensionais dotados de funções de cálculo, parametrização e *rendering* que propiciam a construção de modelos com aparência realística. A seguir serão descritas as principais características dessas ferramentas:

- a) Editores de Desenho: Contém ferramentas para criar e modificar imagens gráficas,

sobre uma base vetorial. Trabalha através do uso de comandos de comunicação homem/máquina (interface gráfica), por meio de diversos menus ou instruções (*prompts*) que aparecem na tela do monitor de vídeo, e equipamentos adequados para que os dados sejam introduzidos e modificados pelo usuário, a exemplo de teclado e *mouse*. As entidades gráficas são definidas através de suas características geométricas, utilizando um sistema cartesiano de coordenadas (x,y,z). No plano, o valor de “z” é sempre igual a zero, enquanto no espaço essa coordenada assume valor positivo ou negativo.

A distância entre duas coordenadas é a medida em unidades do desenho. O programa entende esta unidade como sendo adimensional, contudo oferece ao usuário opções de sistemas de unidades variados, a exemplo de: decimal, científica, métrico, polegadas, etc incluindo também diferentes tipos de medidas de ângulos (graus, grados e outros). Sendo assim, a unidade de trabalho é escolhida pelo usuário, cujos desenhos são feitos, em geral, em escala real, considerando uma área de desenho infinita, ou seja, a área de desenho pode ser ampliada indefinidamente, através de recursos de visualização. A escala do desenho só será definida na hora de traçá-lo no *plotter* ou impressora.

Uma vez criados, os objetos do desenho podem ser manipulados através de recursos de edição que possibilitam, entre outras coisas:

1. Deslocamentos (*Move*);
2. Rotações (*Rotate*);
3. Simetrias ou espelhamento (*Mirror*);
4. Cópia simples (*Copy*) e cópias múltiplas (*Array*);
5. Mudança de escala (*Scale*);
6. Esticamento (*Stretch*), prolongamentos (*Extend*), encurtamentos (*Trim*) e rompimentos (*Break*);
7. Chanfros (*Chamfer*) e concordâncias (*Fillet*);

Os editores de desenho possuem também recursos para colocar automaticamente as dimensões ou cotas dos desenhos. Os tipos básicos de cotas são: linear, angular, diâmetro e raio, que podem ser configuradas para atender a diferentes unidades e escalas em um mesmo desenho, bem como alterar a sua aparência. Permitem realizar mensurações e cálculo de

distâncias, áreas e perímetros, entre outros.

Desenhos de certa complexidade exigem critérios de organização para que se possa ter um total controle de suas partes. A divisão dos desenhos em camadas (*Layers*) permite a organização do processo de trabalho de forma bastante eficaz. As camadas de trabalho são semelhantes a folhas transparentes de papel, em que cada folha carrega uma parte do desenho, que, sobrepostas, formam o todo. Estas camadas são autônomas e podem posicionar todos os objetos de interesse, identificadas através de nomes ou números determinados pelo usuário, a cada uma delas deve corresponder uma cor e um tipo de linha. As cores atribuídas aos *layers* é que, posteriormente, irão determinar as espessuras do traçado para a plotagem. Estas camadas de desenho podem estar invisíveis ou visíveis no monitor do computador, ou mesmo visíveis, mas não disponíveis para qualquer alteração nas entidades nelas contidas (congeladas).

Bibliotecas de elementos gráficos é um recurso utilizado para redução do tempo de desenho, onde é possível armazenar elementos padrões e detalhes que podem ser reproduzidos em projetos diversos. Estes elementos são armazenados na forma de blocos, ao qual é associado um nome, que é utilizado como elemento de inferência para sua inserção no desenho. De forma geral, um bloco pode ser composto de várias entidades com vários níveis de trabalho, várias cores e vários tipos de linha, mas é tratado como uma entidade única, quando inserida em uma camada de desenho. Os editores de desenho de última geração permitem, também, a construção de blocos parametrizados. Informações textuais podem ser associadas a estes blocos constituindo atributos, agindo como uma ligação entre a informação gráfica do desenho e uma base complementar de dados, externa a ele. Esse recurso é muito utilizado para inserir desenhos que se repetem, mas que devem carregar informações textuais diferentes a cada inserção, a exemplo do carimbo das pranchas.

Algumas funcionalidades adicionais constituem ferramentas de ajuda "transparentes" às funções de desenho e edição, ou seja, que ao serem ativadas não interrompem as outras.

Algumas destas funcionalidades são:

- Disponibilidade de *grids*, malhas, possibilitando trabalhar com módulos adequados a cada projeto;
- Modo de seleção em pontos notáveis (ponto médio, tangente, ponto final, perpendicular etc.), permitem a realização das operações com maior precisão;



- Modo de visualização também são elementos de ajuda importantes, pois, através deles, se pode visualizar o conjunto do desenho, ou uma porção onde se pretende trabalhar, auxiliando na seleção de partes do desenho e conferindo maior rapidez nas operações;
- Diferentes sistemas de coordenadas também constituem elementos importantes no auxílio ao desenho.

b) Modeladores Geométricos Tridimensionais: Enquanto os editores de desenho são aplicações que substituem o ferramental tradicional, os modeladores tridimensionais, também conhecidos como modeladores geométricos tridimensionais, permitem a construção de modelos matemáticos dos objetos, que podem ser visualizados de diversas formas, estática e dinamicamente. Os métodos correntes mais comuns de geração de modelos tridimensionais utilizam abordagens diferentes por meio da geometria para construí-los. São eles: CSG (*Constructive Solid Geometry*), que utiliza sólidos primitivos, cilindros, paralelepípedos, cones e esferas, combinados através de operações booleanas (união, diferença e interseção), para construir o objeto final; e B-Rep (*Boundary Representation*), que utiliza superfícies (faces) conectadas para construir o objeto. Estes dois métodos são frequentemente combinados para facilitar a geração de modelos complexos, estando presentes na maioria dos modeladores geométricos de uso comercial, configurando um sistema híbrido de modelamento. As ferramentas de modelagem geométrica tridimensional apresentam três tipos básicos de modelos:

1. *Wireframe* - também chamado de modelamento de arame, ou modelo aramado, Pode ser comparado a um modelo físico confeccionado com fios de arame. Representa o objeto tridimensional apenas pelos seus vértices e arestas, apresentando uma malha de linhas que permite visualizar o contorno do modelo;
2. Superfícies - Através dos pontos definidos pela malha (*wireframe*), definem matematicamente as superfícies que formam o contorno do modelo. Assim o modelo de superfície considera não só os vértices e arestas do modelo, mas os planos formados por eles;
3. Sólidos - Representação matemática completa do objeto, definido como um volume sólido, capaz de reconhecer as propriedades geométricas do modelo.

Objetos tridimensionais simples também podem ser construídos através de operações de extrusão e revolução de formas primitivas, entretanto a modelagem de formas complexas

requer transformações sucessivas a partir de formas básicas. Os modelos geométricos tridimensionais podem ser submetidos a uma vasta gama de análises no sentido de verificação de suas características físicas e geométricas, representando um avanço no processo de projeto em benefício de sua qualidade. Remete a um processo de projeto mais próximo da forma "natural" do arquiteto trabalhar, gerando inúmeras possibilidades de apropriação de informações, o que propicia uma abordagem do projeto mais precisa, através de recursos como:

- Visualização da futura obra em várias fases de construção;
- Simulação do processo construtivo, dos materiais de construção e acabamento;
- Visualização do modelo em várias escalas e níveis de detalhamento;
- Deslocamento do usuário em torno do modelo, através de recursos como *walk around* e *walk throug*.

Algumas ferramentas de modelamento geométrico tridimensional permitem a obtenção automática de desenhos (vistas ortográficas) a partir da inferência de secções no modelo 3D. Todos os desenhos extraídos de um mesmo modelo têm a garantia de estarem em conformidade um com o outro. Uma geração de modeladores mais recentes introduziram a modelagem paramétrica. Nestas ferramentas, a forma é definida de maneira semelhante às demais descrita anteriormente, mas em adição à sua geração, o modelador retém o conjunto de *inputs* (entradas) da operação. Essas operações formam um conjunto de operadores e operandos na forma de expressões algébricas, as quais podem ser posteriormente editadas, graficamente ou alterando-se suas variáveis, e a forma automaticamente regenerada com sua nova configuração. Esta capacidade de edição é muito útil em projeto, permitindo operações de edição de formas pela mudança de seus parâmetros, ou efetuando operações em um modelo pré-existente, recurso muito útil para a atividade projetual onde a geometria do produto é refinada constantemente. Embora ainda não seja possível lançar mão do modelo geométrico tridimensional para automatizar a construção de um objeto arquitetônico, como se faz em algumas áreas da engenharia, além de sua utilidade como dado de entrada para diversos tipos de análises de ordem técnica, o modelo tridimensional é frequentemente utilizado por arquitetos e projetistas como meio de facilitar a compreensão do seu trabalho por parte de leigos.

c) Acabamento e *Rendering* - Sendo o modelo tridimensional a descrição do objeto em uma estrutura de dados espaciais, a renderização é o processo pelo qual se pode obter imagens digitais do tipo *raster* extraídas deste modelo. Neste processo trabalha-se com ferramentas que simulam a aplicação de texturas no objeto, iluminação "natural" e artificial, em busca de um resultado foto-realístico, ou seja, simulando ambiente e estruturas verossímeis, obtidos por meio de técnicas de *raytracing*<sup>54</sup>, dando a aparência final do objeto, determinando-se também posições do observador em relação ao objeto, ou do objeto em relação à câmera, para determinar o ângulo de visão em que será sintetizada a imagem. Como produto, uma grande variedade de *renders* estão disponíveis, alguns integrados em pacotes de *software* de modelagem e animação, outros são *stand alone*, ou seja, operam independentes, mas para isto é preciso que "leiam" os arquivos provenientes do modelador.

Uma imagem renderizada pode ser entendida em termos do número de feições visíveis, sejam elas: sombras, degradês, texturas, cores, iluminação, transparências, reflexos, etc. As diferentes opções de aplicação de texturas nos objetos ficam armazenadas em "bibliotecas", que são fornecidas juntamente com o programa, e podem ser ampliadas pela modificação das propriedades dos materiais existentes, ou pela inclusão de novos materiais, fornecidos por terceiros. Podem também ser inclusos: vegetação, figuras humanas, figuras de animais, automóveis e fotografias. Muitos algoritmos de renderização têm sido pesquisados, e os *softwares* utilizados podem empregar diferentes técnicas para obter uma imagem final. A grande maioria dos *softwares* de modelamento tridimensional, hoje em dia, trazem o aplicativo de renderização agregado, entretanto o bom resultado depende do conhecimento das técnicas de renderização e da experiência do usuário.

### **Aplicações de Simulação**

Kalisperis (1996), define as aplicações de simulação como aquelas em que ferramentas computacionais são utilizadas para testar e prever a performance do edifício, ou parte dele, concernente às restrições e condicionantes de desempenho estabelecidas, considerando-se o uso a que se destina e o ambiente no qual será inserido.

A integração de aplicações CAD com outras tecnologias, o aprimoramento de técnicas e

---

<sup>54</sup> *Raytracing*, ou traçado de raios é um algoritmo para síntese de imagens digitais.

procedimentos, a exemplo do Método de Elementos Finitos (MEF), permitiu que os modelos matemáticos gerados em ferramentas CAD pudessem servir como dado de entrada para programas destinados a desempenhar análise e simulações, automatizando tarefas comuns a vários ramos da engenharia (MAZZA, 2000). O CAE (*Computer Aided Engineering*) consiste em um sistema para análise de modelos criados em programas CAD, trazendo uma enorme mudança em áreas específicas, tais como a de projeto estrutural, economia de energia em edificações, projeto do canteiro e do processo construtivo. As ferramentas disponíveis e os tipos de simulação possíveis são os mais variados. Para citar as mais difundidas, buscou-se em Amorim (2000) os seguintes exemplos de aplicações de simulação:

- resistência a esforços mecânicos;
- condutividade térmica;
- comportamento acústico;
- estudos de iluminação e sombreamento;
- análises funcionais;
- estudos de volumetria e de interferências.

### **Aplicações de Geração e Otimização**

Ao considerar aplicações de geração e otimização, Kalisperis (1996) deixa claro que essas aplicações estão intimamente ligadas, uma vez que refere-se às ferramentas computadorizadas voltadas para a geração automática de soluções de projeto, onde, em um senso rigorosamente matemático, a otimização significa a obtenção da melhor solução para o problema. Conforme pontuado anteriormente, as tentativas de utilizar o computador para geração de soluções de projeto remontam à própria gênese da aplicação dos computadores em arquitetura, encontrando uma linha de desenvolvimento na integração da tecnologia CAD com técnicas de Realidade Virtual e Sistemas Especialistas. Contudo, para Kalisperis (1996), as aplicações computacionais para geração e otimização de soluções de projeto ainda são muito limitadas, decorrente do caráter aberto do problema arquitetônico, que inclui variáveis atreladas à cognição, experiência, intuição e criatividade do projetista. Não distante dessa visão, Amorim (2000) admite que, através das tecnologias disponíveis, é possível o desenvolvimento de "sistemas inteligentes de criação", entretanto, destinam-se a [...] *aplicações especializadas e limitadas, caracterizadas por um universo restrito e fechado, fundamentado por uma base de*

*dados já validada de problemas e soluções*, (AMORIM, 2000, p.11). Segundo Kalisperis (1996), essas aplicações são utilizadas, principalmente, no planejamento do *layout* de edifícios e em aspectos específicos do projeto, como estrutura, conforto térmico e execução da obra.

As ferramentas CAD se difundiram com muita ênfase a partir dos anos 1980, e hoje constituem um tema bastante amplo, pois nelas se inserem continuamente novos recursos, utilizados na produção de projetos dos mais variados. Vale ressaltar que as aplicações aqui apresentadas perfazem o conjunto básico de funcionalidades de um sistema computadorizado de auxílio ao projeto de arquitetura, existindo uma série de recursos mais ou menos avançados que auxiliam e que complementam o processo de projeto, como ferramentas de gestão e controle do fluxo de informações, planejamento e execução da obra, os Sistemas Gerenciadores de Documentação Eletrônica (EDMS)<sup>55</sup>, os geradores de fluxogramas, as conhecidas planilhas eletrônicas ou simples rotinas de cálculo (AMORIM, 2000).

## 2.8 Tendências

De acordo com Eastman (1999) a evolução do CAD está estritamente ligada ao desenvolvimento das seguintes tecnologias:

- Tecnologia dos processadores de dados
- Tecnologia dos monitores de vídeo
- *Software* e algoritmos

Como resultado do desenvolvimento dessas tecnologias a tendência atual é de que ferramentas CAD integradas agrupem, cada vez mais, múltiplas aplicações em torno de uma única representação do edifício, (NAVEIRO; BORGES, 1998). Eastman (1999) considera, ainda, que o estado da arte foi determinado por um processo evolutivo onde destacam-se três eixos básicos de desenvolvimento da tecnologia CAD:

- CAD como Editor Geométrico: enfatiza os benefícios da transição da representação 2D para a 3D, destacando-se entre eles: diversidade de formas de visualização,

---

<sup>55</sup> *Eletronic Document Management System* no idioma inglês.

automação de máquinas e manufaturas, obtenção de consistência total entre as representações bidimensional e tridimensional - uma vez que a primeira passou a ser uma das saídas da segunda, entrada para vários tipos de análise e a modelagem paramétrica,

- CAD como Plataforma de Desenvolvimento de Aplicações: essa é uma funcionalidade incorporada aos sistemas CAD no início dos anos 1990, graças, principalmente, aos avanços no desenvolvimento das técnicas de programação orientada a objetos e à característica modular da estrutura desses programas. A introdução dessa funcionalidade dotou os sistemas CAD de capacidade de customização<sup>56</sup>, permitindo que funcionalidades específicas voltadas para determinadas áreas do conhecimento, como arquitetura e engenharia, fossem incorporadas a sistemas de uso genérico, automatizando um número maior de tarefas. A incorporação de capacidades específicas é que fez com que o CAD pudesse ser visto como uma verdadeira ferramenta de auxílio ao projeto (EASMAN, 1999).
- Modelo do Edifício: Eastman (1999) ressalta que os primeiros esforços rumo a essa mudança de paradigma iniciam-se no final dos anos 1960 e decorrem dos anos 1970, em centros de pesquisa de Universidades dos Estados Unidos da América e do Reino Unido. A hipótese sustentada era de que a tarefa básica do projeto é desenvolver as especificações do edifício e analisar o comportamento de seus componentes no ambiente da composição. Para atender a essa prerrogativa, esses sistemas baseavam-se no uso do editor geométrico tridimensional de sólidos, integrado a uma base de dados relacionais para o armazenamento de dados não geométricos. Segundo Eastman (1999), os esforços empreendidos definiram uma única e coerente representação do edifício, em torno da qual aplicações específicas foram construídas. Contudo, as limitações naquela época eram muito grandes face ao então estágio de desenvolvimento de *hardware*, linguagens e técnicas apropriadas de programação e na própria forma de estruturar os programas CAD.

Como resultado dos avanços alcançados no âmbito dos três eixos básicos, acima referidos, de desenvolvimento da tecnologia CAD, no campo específico do projeto do edifício, a idéia predominante é de que ferramentas computacionais específicas, necessárias em diferentes estágios para o auxílio ao projeto, e algumas ferramentas de apoio à construção, possam ser

---

<sup>56</sup> Adaptação, personalização, modificação de um sistema para satisfazer os requerimentos do usuário.

integradas. Essa tecnologia emergente está sendo denominada de BIM (*Building Information Model*), tratando-se de uma representação digital que pretende incorporar, através de um único modelo geométrico tridimensional, informações de todas os elementos integrantes da edificação, capturando a forma, o ambiente e as relações entre as partes, envolvendo as várias atividades relacionadas com a progressão do projeto, além do completo ciclo de vida do edifício (EASTMAN, 1999). Conseqüentemente, conceitos, análises, produção, *marketing*, eficácia, durabilidade e outros aspectos, deverão ser considerados durante o processo projetual.

A grande diversidade de tecnologias que podem ser aplicadas no projeto do edifício e, similarmente, as diferentes formas que podem ser desenvolvidas, remetem a diferentes componentes, regras de composição e, conseqüentemente, a diferentes performances e métodos de análise. Assim o BIM está encontrando, em primeira instância, campo de desenvolvimento na indústria aeroespacial, automobilística e de produtos manufaturados (EASTMAN, 1999). Eastman (1999), pondera que isto ocorre porque tais indústrias trabalham com produtos onde é frequente a possibilidade de identificar o conjunto de seus componentes e os atributos necessários para indicar a sua forma física, possíveis composições, funções e performances. O mesmo não ocorre com a indústria da construção, onde o projetista é livre para optar entre uma variedade enorme de sistemas construtivos, sistemas de instalações, revestimentos, *layouts*, esquadrias, e assim por diante. Alguns desenvolvedores de CAD, líderes de mercado, como a Autodesk, têm procurado agregar algumas dos conceitos e capacidades do BIM a seus produtos, a exemplo do AutoCAD Architecture e o Revit (AUTDESK, c2010).

Para Krygiel e Nies (2008), em AEC, BIM é mais do que uma aplicação, é uma troca de metodologia de produção da documentação na indústria do projeto e construção. É informação sobre todo o edifício e um completo conjunto de documentação armazenada em uma base de dados. Todas as informações são paramétricas e interconectadas. Segundo os autores, essa transição decorre também da necessidade de se equacionar a forma de pensar e produzir arquitetura face ao aumento da complexidade dos edifícios, que incorporaram elementos da própria tecnologia computacional e das telecomunicações em seus sistemas de instalações.

A tendência de desenvolvimento da tecnologia computacional aplicada ao projeto envolve

uma única representação digital da edificação, usada com praticidade durante os processos de projeto, fabricação, e nas operações de manutenção do edifício, possibilitando que os agentes envolvidos nestes processos possam usufruir das facilidades do uso de ferramentas baseadas na computação. Espera-se que esta mudança afete todo o mercado imobiliário, a forma de negociação do setor e a organização da indústria da construção, além da formação de profissionais de arquitetura e engenharia. Fatalmente irá ocorrer esta transição, mas ainda é preciso desenvolver alguns métodos e estratégias para superar limitações técnicas e organizacionais, que provavelmente não serão resolvidas rapidamente.

Diante do exposto, obteve-se um panorama geral do desenvolvimento do aparato computacional empregado na atividade projetual, identificando-se as principais tecnologias e ferramentas desenvolvidas para essa finalidade, assim como as tendências dessa automação. Baseado nesse conhecimento é que se deu prosseguimento ao desenvolvimento do próximo capítulo, onde buscou-se identificar programas livres gráficos com as características e os atributos requeridos, frente ao contexto atual de automação do projeto.



**3. AVALIAÇÃO DE SOFTWARE:  
programas livres gráficos utilizados na Arquitetura**

### 3.1 Normas de Avaliação de de *Software*

Esse capítulo tem o objetivo de estabelecer procedimentos metodológicos para se chegar a um conjunto de *softwares* livres gráficos com potencial de aplicação no desenvolvimento de projetos de arquitetura, bem como elaborar, com base em métodos cientificamente aceitos, um processo para avaliação qualitativa desses programas, quando aplicados a uma experiência prática da atividade projetual. Como resultado desse processo, reuniu-se um conjunto de informações que permitiram a formulação de conclusões acerca das perspectivas de uso dessas ferramentas na arquitetura, bem como fazer recomendações para o desenvolvimentos de trabalhos futuros relacionados com esse tema.

Dentro desse contexto, buscou-se na literatura especializada, e no arcabouço normativo da ABNT (Associação Brasileira de Normas Técnicas)<sup>57</sup>, técnicas e procedimentos para realização de avaliação de produto de *software*, estabelecendo, dessa forma, uma linha de raciocínio lógico para desenvolvimento da metodologia a ser empregada.

A avaliação de programas de computador está relacionada com a medição de sua qualidade (REIS, 2003; TELES, 2005), entendendo-se qualidade como a totalidade das características que confere ao produto a capacidade de satisfazer às necessidades, explícitas e implícitas, em um determinado contexto de uso. A ABNT dispõe de um conjunto de normas, elaboradas com base em prescrições normativas geradas pela ISO/IEC (*International Organization for Standardization* e *International Eletrotechnical Comission*)<sup>58</sup>, que alicerçam técnicas e procedimentos aplicados no processo de avaliação de ferramentas computacionais (TELES, 2005).

Além dos métodos e técnicas indicados nesse conjunto de normas, existem outros, mais ou menos difundidos, que consideram a avaliação do ponto de vista da qualidade. Entre as mais difundidas está a NBR ISO/IEC 25051, denominada de modelo SQuaRE (*Software Quality*

---

<sup>57</sup> O conjunto de normas utilizados constam em (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2003a; 2009; 2001a; 2003b; 2003c; 2001b; 2004).

<sup>58</sup> A *International Organization for Standardization*, ou, Organização Internacional para Padronização, traduzido para o idioma português, é uma rede de instituição de padrões nacionais, formada por 162 países, cuja central de coordenação está sediada em Genebra na Suíça. Esses padrões são estabelecidos com base em uma avaliação consensual de comitês de especialistas, que no campo da eletrotécnica constitui-se no âmbito da *International Eletrotechnical Comission*, ou Comissão Internacional de Eletrotécnica, traduzido para o idioma português.

*Requirements and Evaluation*)<sup>59</sup> aplicável ao chamado produto de *software* comercial “de prateleira”, a exemplo do Office da Microsoft; o CMMI (*Capability Maturity Model Integration*)<sup>60</sup> (RINCON, 2009) e o GQM (*Goal Question Metric*)<sup>61</sup> (MARIA, 2010). Existe ainda a ABNT (2009) voltada para o estabelecimento de critérios de qualidade aos processos de todo o ciclo de vida do programa. Todas essas normas são correlatas e, por vezes, complementares, na medida em que tratam da questão da avaliação da qualidade do processo e do produto. Supondo-se que uma coisa depende da outra, podemos afirmar que a qualidade do produto depende da qualidade do processo, portanto, o resultado da avaliação da qualidade do produto fornece dados que podem ser utilizados para a melhoria do processo.

Como visto no capítulo 1, o processo de desenvolvimento do *software* livre não está necessariamente atrelado a um modelo formal de estabelecimento de requisitos de qualidade, tão pouco a um processo institucionalizado de avaliação e, nesse contexto, a garantia do controle de qualidade varia de acordo com o nível de comprometimento do mantenedor ou grupo de mantenedores do projeto, apoiados em uma política de revisão e integração do código, e no *feed back* dos usuários (REIS, 2003).

Para os fins deste trabalho não cabe a avaliação da qualidade do processo do ciclo de vida dos programas gráficos livres que serão estudados, porque não está em questionamento o processo de desenvolvimento, distribuição e manutenção dessas ferramentas, mas o que se pretende é chegar a conclusões relacionadas com a aplicação prática delas no desenvolvimento de projetos de arquitetura, portanto, o foco está na utilização do produto.

Dessa forma, a construção da estrutura do processo de seleção e da avaliação dessas ferramentas irá se apoiar no conjunto de normas apresentadas no quadro 4. Observando-se esse quadro, percebe-se que a série NBR ISO/IEC 9126 alicerça a formatação de um modelo de qualidade e apresenta métodos e escalas para medição de atributos (ABNT, 2003a), enquanto a série NBR ISO/IEC 14598 fornece requisitos e recomendações para o procedimento da avaliação, considerando três grupos distintos de público: desenvolvedores, adquirentes e avaliadores (ABNT, 2001a). Antes de se lançar mão dessas normas para

---

<sup>59</sup> Requerimentos e Avaliação de Qualidade de Software, traduzido para o idioma português.

<sup>60</sup> Modelo Integrado de Maturidade da Capacidade, traduzido para o idioma português.

<sup>61</sup> Objetivo – Perguntas – Métricas, traduzido para o idioma português. Baseia-se no fato de que deve existir uma necessidade clara associada a cada métrica.

formatação da metodologia de seleção dos programas e execução do experimento prático, será feito um breve apanhado dos conceitos, técnicas e procedimentos nelas previstos, buscando o bom entendimento de seu conteúdo e da interface existente entre as partes que compõem cada uma delas.

**Quadro 4** – Série de normas da ABNT para avaliação de produto de *software*.

<b>NORMA</b>	<b>PARTES</b>	<b>OBJETIVO</b>
NBR ISO/IEC 9126 “Engenharia de Software – Qualidade de produto”	Parte 1: Modelo de qualidade	Descrever um modelo de qualidade do produto de <i>software</i> .
	Parte 2: Métricas externas (versão português em elaboração)	Apresentar métodos e escalas de medição de atributos externos.
	Parte 3: Métricas internas (versão português em elaboração)	Apresentar métodos e escalas para medição de atributos internos.
	Parte 4: Métricas de qualidade em uso (versão português em elaboração)	Apresentar métodos e escalas para medição dos efeitos do uso do <i>software</i> , em condições específicas.
NBR ISO/IEC 14598 “Engenharia de Software – Avaliação de produto”	Parte 1: Visão Geral	Fornecer uma visão geral das outras partes e explica o relacionamento entre NBR ISO/IEC 14598 e o modelo de qualidade apresentado na NBR ISO/IEC 9126.
	Parte 2: Planejamento e Gestão	Fornecer requisitos, recomendações e diretrizes para o planejamento e a gestão da avaliação de produto de <i>software</i> .
	Parte 3: Processo para desenvolvedores	Fornecer requisitos e recomendações para a implementação da avaliação quando ela é conduzida em paralelo com o desenvolvimento e executada pelo desenvolvedor.
	Parte 4: Processo para adquirentes	Fornecer requisitos, recomendações e orientações para a medição, julgamento e avaliação sistemática da qualidade de produto de <i>software</i> visando a aquisição ou modificação do produto.
	Parte 5: Processo para avaliadores	Fornecer requisitos e recomendações para implementação prática da avaliação, destinada à aqueles que a executam de forma independente, por solicitação de desenvolvedores, adquirentes ou outra parte interessada.
	Parte 6: Documentação de módulos de avaliação	Definir a estrutura e o conteúdo da documentação a ser usada durante o processo de avaliação.

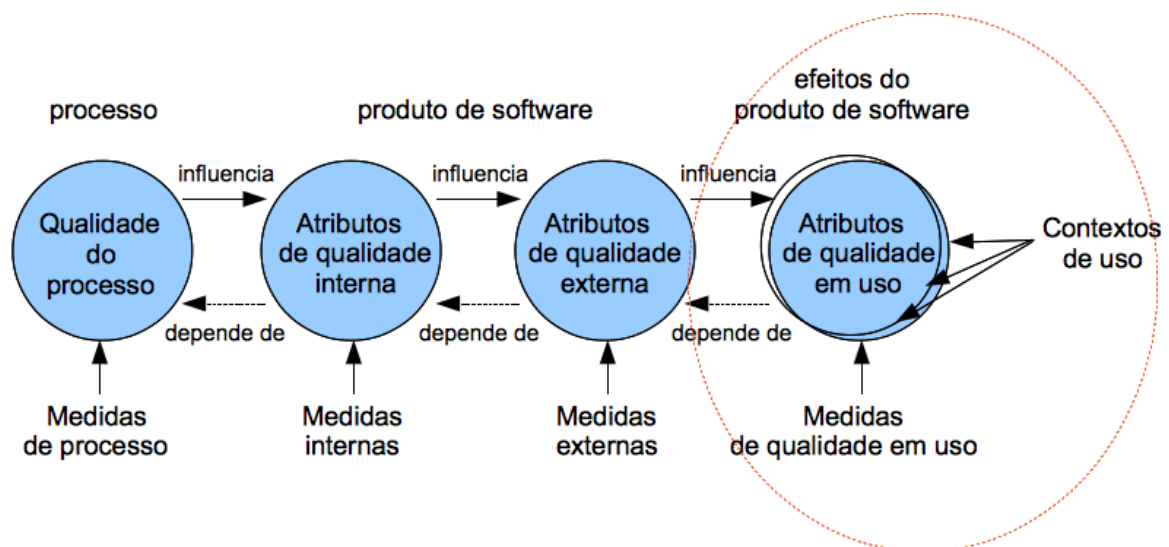
Fonte: – Elaborado pelo autor com base em (ABNT, 2001a).

### 3.2 O Modelo de Qualidade da NBR ISO/IEC 9126

A NBR ISO/IEC 9126 define três abordagens para a qualidade do produto de *software*:

1. A qualidade interna: relacionada com os atributos de qualidade, durante a implementação do código-fonte (produto intermediário);
2. A qualidade externa: relacionada com os atributos de qualidade quando o programa é integrado a um sistema computacional, ou seja, quando ele é executado (ainda em fase de testes);
3. A qualidade em uso: relacionada com os atributos de qualidade, dentro da perspectiva de satisfação do usuário, quando o produto é utilizado em um contexto especificado.

Sendo assim, essas abordagens definem três instâncias de avaliação que fazem parte dos processos no ciclo de vida do *software*. Seguindo esse modelo, a qualidade final da ferramenta deriva da qualidade dos produtos intermediários gerados no processo de seu desenvolvimento, que pode ser avaliada medindo-se seus atributos de qualidade interna. Baseado no conhecimento da qualidade interna, pode-se prever a qualidade para o produto final, que é a sua qualidade externa, tipicamente avaliada quando o programa está sendo testado em um ambiente simulado. Dessa forma, o modelo de qualidade definido nessa norma é composto de duas partes; a primeira descreve um modelo para as qualidades interna e externa, e a segunda, um modelo para a qualidade em uso (MARIA, 2010).



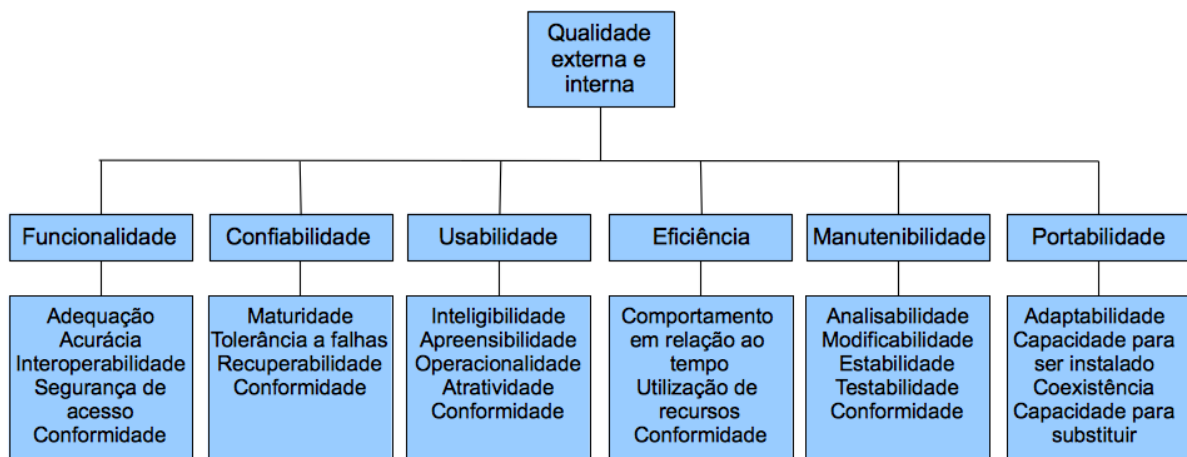
**Figura 6** – Qualidade no ciclo de vida do *software*

Fonte: Elaborado pelo autor com base em (ABNT, 2003a, p. 4)

A figura 6 ilustra a abordagem de interdependência qualitativa nas diferentes fases de desenvolvimento de um programa de computador, ressaltando-se que, como visto anteriormente, a avaliação de qualidade dos processos no ciclo de vida do *software* encontra-se prescrita em ABNT (2009) e, no âmbito da abordagem da ABNT (2003a) o enfoque deste trabalho está na qualidade em uso, conforme destacado na figura 6.

Para medir a qualidades externa e interna, o modelo definido pela NBR ISO/IEC 9126 decompõe hierarquicamente os atributos de qualidade em características e subcaracterísticas (figura 7). Essas subcaracterísticas podem ser medidas por meio de métricas externas e internas, respectivamente exemplificadas nas partes 2 e 3 da norma.

Os quadros I a VI, constantes do APÊNDICE B, apresentam, de forma sintetizada, as definições atribuídas a cada característica e subcaracterística do *software* que influencia sua qualidade, de acordo com o modelo especificado na parte 1 da NBR ISO/IEC 9126.



**Figura 7** – Modelo de qualidade para qualidade externa e interna.

Fonte: (ABNT, 2003a, p 7.).

O modelo definido para as qualidades interna e externa define uma subcaracterística de conformidade para todas as características (ABNT, 2003a), visando a não contrariedade a normas convenções e regulamentações que, por ventura, estejam a elas relacionados, já que se trata de um modelo genérico, aplicável a uma diversidade de contextos de uso. Segundo Teles (2005), o desdobramento das características em subcaracterísticas ajuda a delimitar a amplitude do universo que a característica engloba, auxiliando na especificação de requisitos de qualidade e dos atributos necessários à satisfação desse requisitos. Por sua vez, a

associação desses atributos a uma escala de valores, de ordem numérica ou qualitativa é que possibilita a medição do nível de qualidade oferecido pelo produto de *software*, cabendo ao avaliador [...] *identificar quais atributos são relevantes no produto que será avaliado*. (AZEVEDO, 2008, p. 17).

O outro modelo de qualidade definido pela NBR ISO/IEC 9126, o de qualidade em uso, classifica quatro características: eficácia, produtividade, segurança e satisfação. Assim, a qualidade em uso pode ser medida avaliando-se o quanto o produto, utilizado por usuários específicos, atende às necessidades desses usuários para que eles atinjam as metas especificadas preservando-se essas características. A importância de se especificar o contexto de uso deriva da diversidade de situações que podem decorrer da utilização de uma mesma ferramenta computacional, sendo que os resultados obtidos com a sua aplicação dependem do nível de conhecimento que o usuário detém sobre sua operação, configuração do sistema ao qual está integrada, do ambiente e da finalidade de sua utilização. O quadro 5 apresenta a definição de cada uma das características de qualidade em uso.

**Quadro 5** – Definição dos atributos de qualidade em uso.

MODELO DE QUALIDADE EM USO	
CARACTERÍSTICA	DEFINIÇÃO
Eficácia	Capacidade do produto de <i>software</i> permitir que usuários atinjam metas especificadas com acurácia e completude, em um contexto de uso especificado.
Produtividade	Capacidade do produto de <i>software</i> permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida, em um contexto de uso especificado.
Segurança	Capacidade do produto de <i>software</i> de apresentar níveis aceitáveis de riscos de danos a pessoas, negócio, <i>software</i> , propriedades ou ambiente, em um contexto de uso especificado.
Satisfação	Capacidade do produto de <i>software</i> em satisfazer usuários, num contexto de uso especificado.

Fonte: Elaborado pelo autor com base em (ABNT, 2003a).

Como pode ser observado no quadro 5, diferente do que se apresenta nos quadros constantes do APÊNDICE B, o modelo de qualidade em uso não desdobra seus atributos além do nível das características, contudo, a norma deixa claro que o alcance da qualidade em uso depende da obtenção da necessária qualidade externa, uma vez que essa última se manifesta quando o

programa é utilizado como parte de um sistema computacional e, dessa forma, a primeira pode ser conferida pela combinação de atributos de qualidade externa referentes à funcionalidade, confiabilidade, usabilidade e eficiência, que influenciam na obtenção da qualidade em uso, possibilitando, assim, uma visão qualitativa, medida em termos do resultado da aplicação da ferramenta em um determinado ambiente, e não de suas propriedades enquanto produto de software (ABNT, 2003a, p.14).

### 3.3 Métricas de Avaliação de Software

A medição não pode ser feita diretamente das características, ou subcaracterísticas, conforme definidas em ABNT (2003a), para isso é necessário estabelecer métricas que se correlacionem às características do programa. Assim, a métrica consiste no método e escala de medição definidos para mensurar o quanto ele atende aos requisitos de uma característica de qualidade específica. Conforme consta do quadro 4, a NBR ISO/IEC 9126 define três tipos de métricas<sup>62</sup>.

- Métrica interna – Aplicada para medir propriedades intrínsecas ao produto de *software* na fase de projeto e codificação. As medições, nesse caso, utilizam números ou frequências de elementos que compõem o programa, e que aparecem, por exemplo, na estrutura do código fonte. Exemplos de métricas internas encontram-se na ISO/IEC 9126-3;
- Métrica externa – Aplicadas na fase de testes do produto de *software*, quando ele está apto a ser executado em um sistema computacional, num ambiente técnico e organizacional especificado. Essas métricas estabelecem limites aceitáveis para quantificar os critérios de qualidade que validam se o programa atenderá às necessidades do usuário, baseado nas suas propriedades, ou atributos de qualidade externa. Exemplos de métrica externa encontram-se na ISO/IEC 9126-2;
- Métricas de qualidade em uso – Valida a qualidade do produto de *software* em cenários de uso específicos. É o efeito combinado das qualidades interna e externa. As medições são feitas a partir dos resultados obtidos com o uso da ferramenta, em um

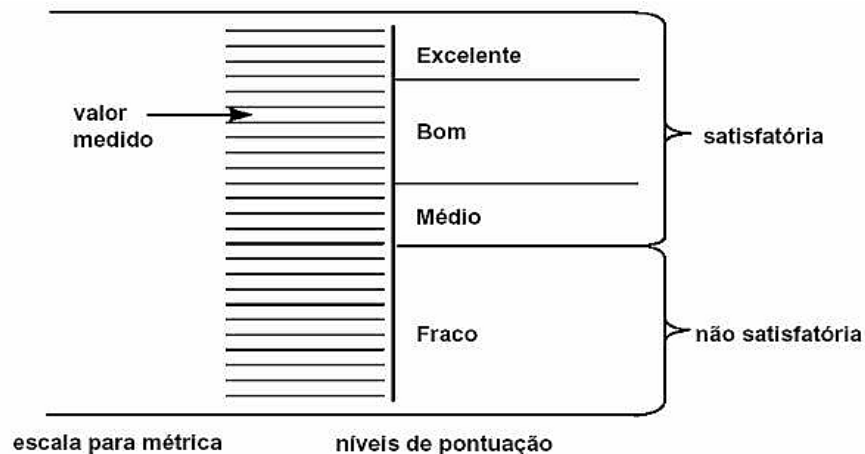
---

<sup>62</sup> A NBR ISO/IEC 9126 define na sua Parte 1 os conceitos de métrica interna, externa e de qualidade em uso, entretanto, as partes 2, 3 e 4, que apresentam respectivamente exemplos de métricas externa, interna e de qualidade em uso, ainda não foram devidamente traduzidas para o português com base nas normas geradas pela ISO/IEC (*International Organization for Standardization*).



contexto especificado, e não de suas propriedades. Exemplos de métricas de qualidade em uso encontram-se na ISO/IEC 9126-4.

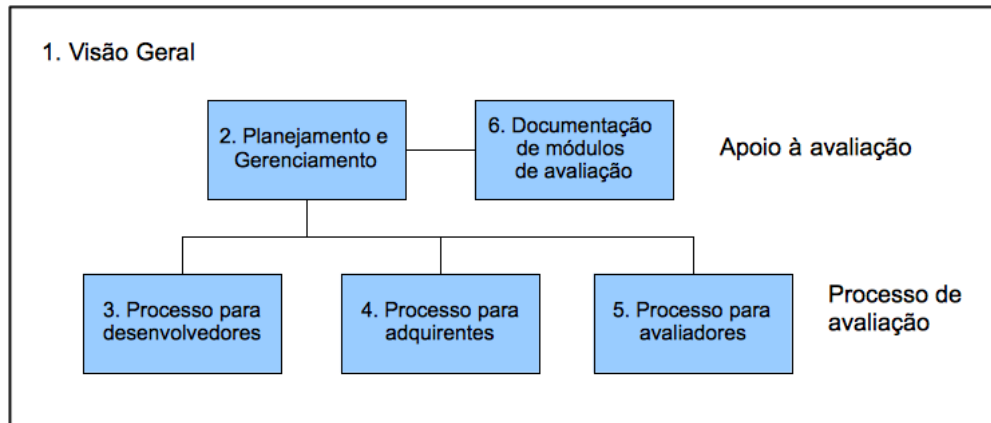
Para aplicação das métricas se faz necessário estabelecer níveis de pontuação dentro de uma escala dividida em faixas, que correspondem a graus de satisfação de atendimento aos requisitos pré-definidos, com base em um modelo de qualidade apropriado e sem ambiguidades (FERREIRA, 2003). A figura 8 ilustra um exemplo de níveis de pontuação para as métricas. O resumo do conjunto dos níveis de pontuação alcançados, para cada uma das características consideradas, resulta na declaração do quanto o programa atende aos requisitos de qualidade.



**Figura 8** – Níveis de pontuação para a métrica.  
Fonte: ABNT (2001a, p.12).

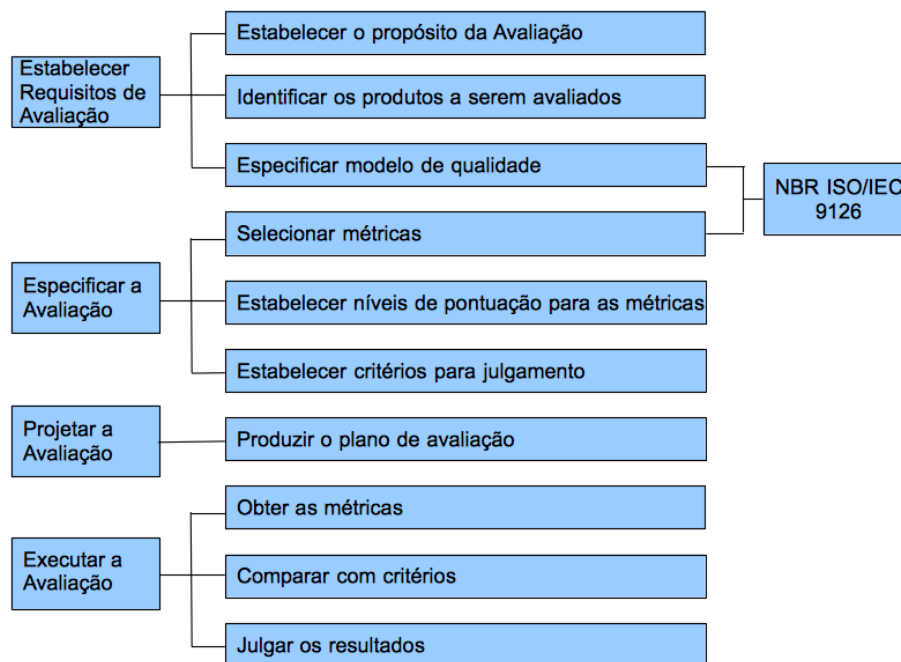
### 3.4 O Método de Avaliação da NBR ISO/IEC 14598

A série de normas NBR ISO/IEC 14598 apresenta métodos para medição, julgamento e avaliação da qualidade de produtos de *software*. Conforme observa-se no quadro 4 apresentado anteriormente, ela é composta de seis partes, cada uma delas com objetivos bem definidos, visando nortear os procedimentos de avaliação de acordo com o público alvo. Enquanto a parte 1 fornece uma visão geral do processo e explica o relacionamento entre a NBR ISO/IEC 14598 e a NBR ISO/IEC 9126, as partes 2 e 6 relacionam-se com o suporte e gestão da avaliação. Já as partes 3, 4 e 5 oferecem, respectivamente, orientação e requisitos em três diferentes situações: desenvolvimento, aquisição e avaliação independente (AZEVEDO, 2008). A figura 9 ilustra o relacionamento entre as partes dessa norma.



**Figura 9** – Relacionamento entre as normas do processo e as de suporte para a avaliação.  
Fonte: ABNT (2001a, p.6).

Ao se proceder a avaliação, as partes 2 e 6 dessa norma prestam orientação para a produção do plano de avaliação e da documentação que conterà o registro das etapas de trabalho, desde o seu planejamento até os resultados e julgamento final. De qualquer forma, de acordo com a norma, sejam quais forem essas circunstâncias, primeiramente deve-se estabelecer os requisitos da avaliação, em seguida especificá-la, depois projetá-la e só então executá-la. A figura 10 mostra um esquema gráfico que representa, de forma geral, essas etapas em todo o processo, os objetivos em cada uma delas, e dentro desses objetivos, aonde se enquadra a utilização do modelo de qualidade definido em ABNT (2003a)



**Figura 10** – Processo de avaliação, segundo a NBR ISO/IEC 14598.  
Fonte: Elaborada pelo autor com base em ABNT (2001a, p.7).

Para realizar essa sequência de ações, o avaliador deverá se apoiar nas técnicas e procedimentos previstos nas partes 3, 4 e 5 da NBR ISO/IEC 14598, optando por uma delas, conforme for o objetivo da avaliação.

### **3.5 Elaboração da Metodologia de Seleção e Avaliação dos Programas Livres Gráficos**

As pretensões deste estudo objetivaram obter resultados que serviram de parâmetro para discernir sobre o quanto os programas gráficos livres selecionados atendiam aos requisitos da atividade projetual especificada, como forma de se estabelecer um juízo acerca do desempenho dessas ferramentas em situações similares à do experimento em uso.

Considerando que a parte 3 da NBR ISO/IEC 14598 introduz prescrições para o processo de avaliação sob a perspectiva do desenvolvedor, automaticamente elimina-se essa opção, porque ela objetiva a medição da qualidade interna do produto, cujos resultados constituem parâmetros para o aperfeiçoamento de seus componentes durante o estágio de desenvolvimento. A parte 4 (Processo para adquirentes), contém orientações para a medição da qualidade do produto de *software* acabado, pronto para execução, visando medir o quanto ele atende aos requisitos estabelecidos para um determinado contexto de uso, com o objetivo de decidir quanto à sua aceitação e conseqüente aquisição. Já a parte 5 (Processo para avaliadores), assume um caráter mais genérico por ser aplicável a todos os processos fundamentais do ciclo de vida da ferramenta, ou seja, pode ser utilizado para avaliar produtos já existentes ou em desenvolvimento, visando aferir conformidade com normas, definir garantia de qualidade e realizar auditorias, para citar alguns exemplos de sua aplicação (ABNT, 2001a). Nesse contexto, a parte 4 da norma é a que se apresenta mais adequada aos propósitos deste estudo, e como tal, será detalhada em seguida.

O processo para adquirentes supõe a realização de avaliação para decisão acerca da aquisição de programas de computador proprietários; pacotes prontos que, normalmente, atendem a determinados segmentos de mercado. Pode também ser aplicado para aquisição de produtos de *software* sob encomenda, desenvolvido para uma determinada aplicação, a partir de uma especificação de requisitos do usuário, ou ainda para indicar modificações em ferramentas existentes. Utilizando uma linguagem genérica, pode-se dizer que, em todos esses casos, a avaliação é aplicada ao produto acabado, pronto para "rodar" sobre um sistema operacional, apto a ser utilizado com determinado propósito, por qualquer um que saiba operá-lo.

Sendo o objeto da avaliação deste trabalho composto por programas gráficos livres, cabe observar que, embora o processo de desenvolvimento dessas ferramentas, e forma de aquisição, sejam dissociados dos padrões adotados pela indústria do *software* comercial, como visto no capítulo 1, ele surge da expectativa de se oferecer alternativas à utilização de produtos comerciais e, como tal, guarda similaridade com os mesmos, ao menos no que se refere à utilização das tecnologias de engenharia de *software* e funcionalidades, podendo ser extremamente variados, tanto quanto os outros.

Dentro desse contexto, reforça-se a justificativa para adoção dos procedimentos prescritos no processo para adquirentes. Surge também como elemento importante para definição da aplicação desse processo de avaliação o fato dessa parte da norma introduzir que: os resultados obtidos podem ser utilizados para [...] *aceitar ou rejeitar um único produto ou, selecionar um produto a partir de várias alternativas, de modo que sejam atendidos os requisitos de qualidade estabelecidos para a aplicação desejada* (ABNT, 2003c, p. 2), fato esse que converge com os objetivos deste trabalho.

Considerando as quatro etapas do processo geral de avaliação de *software*, representadas anteriormente na figura 10, essa parte da norma implementa e refina essas etapas para focar especificamente na avaliação da qualidade do produto final.

Os itens 6 – *Avaliação na aquisição de produtos de software de prateleira* (ABNT, 2003c, p. 7) e 7 – *Avaliação durante a aquisição de software sob encomenda e modificação em software existente* (ABNT, 2003c, p.13), constantes do corpo da norma, apresentam o detalhamento dessas etapas de forma genérica para cada uma das situações, igualmente genéricas, nelas previstas, de forma que, permite-se ao avaliador, selecionar entre os procedimentos ali apontados, quais são aqueles que se enquadram no contexto da avaliação de um produto de *software* qualquer, a ser adquirido para uma determinada finalidade. Esse aspecto confere grande flexibilidade na aplicação do método prescrito no documento normativo e, com base nessa propriedade, foi possível construir o quadro 6, que representa uma síntese da metodologia a ser aplicada no processo de avaliação, no âmbito deste trabalho, contendo as etapas, tarefas-chave, entradas e saídas a serem consideradas, similarmente à Tabela 1 – Processo de avaliação na aquisição de produtos de prateleira, disponível em ABNT (2003c, p.7).

**Quadro 6** – Síntese do processo de avaliação do "pacote" de programas gráficos livres.

ENTRADA	ETAPA	TAREFAS-CHAVE	SAÍDAS
Requisitos de sistema e de <i>software</i>	Estabelecer requisitos de avaliação	Estabelecer o conjunto de requisitos de qualidade do <i>software</i> com base na NBR ISO/IEC 9126-1	Especificação de requisitos de avaliação
		Estabelecer abrangência, propósitos e objetivos da avaliação com relação a: aplicação do <i>software</i> ; perfil do usuário; ambiente da avaliação; tarefas a serem executadas; formato de dados; tratamento de erro ou falha; questões relativa ao tempo e restrições de gestão da avaliação	
Requisitos de avaliação	Especificar a avaliação	Identificação das métricas de qualidade em uso <sup>63</sup> .	Especificação da avaliação
		Estabelecer o nível de cobertura dos requisitos de avaliação	
		Identificar os dados a serem coletados e analisados	
		Especificar os métodos de avaliação específicos a serem usados	
Especificação da avaliação	Projetar a avaliação	Especificar fontes de acesso à documentação, equipamentos a serem utilizados, cursos e/ou treinamentos necessários.	Plano de avaliação
		Especificar o conhecimento requerido pelo avaliador para conduzir a avaliação	
		Especificar a responsabilidade pela execução da avaliação e o cronograma requerido	
		Identificar limitações, ou deficiências, no método de avaliação	
		Especificar os recursos requeridos	
		Identificar pontos de decisão no processo de avaliação que determinam quando, e porque, a avaliação seria considerada completa e encerrada (critérios de aceitação e rejeição)	
		Para cada item da atividade de avaliação especificar: procedimentos e técnicas a serem seguidos, informações de entrada e saída e documentação requerida, requisitos de formato e conteúdo sobre qualquer documentação produzida.	
		Especificar os procedimentos para desenvolvimento e validação das métricas e para padronização do processo de avaliação.	
Plano de avaliação	Executar a avaliação	Documentar e analisar a avaliação para determinar a abrangência de deficiências específicas, limitações, condições especiais de uso do <i>software</i> e necessidade de avaliações adicionais.	Resultados e registro da avaliação
		Registrar a execução, passo-a-passo, dos procedimentos de avaliação, resoluções ou soluções de contorno na eventualidade de uma deficiência.	
		Elaborar conclusões com respeito à possibilidade de aceitação do produto.	

Fonte: Elaborada pelo autor com base em ABNT (2003c, p. 7).

<sup>63</sup> Cabe ao usuário identificar os valores limites reais de aceitação, baseado em experiência, já que não há regras rígidas e fixas para alguns desses valores, (ABNT, 2003c, p. 9).

Na montagem do quadro 6, não se considerou as tarefas relacionadas com identificação e especificação dos seguintes fatores constantes do detalhamento desta parte da norma:

- Nível de integridade – referente à manutenção de riscos do sistema dentro de limites aceitáveis, para itens cujas falhas possam representar algum tipo de ameaça à integridade de ambientes e pessoas, como o caso de sistemas de segurança;
- Custos – referente ao aporte financeiro envolvido para realização da avaliação e aquisição da ferramenta;
- Avaliações anteriores feitas por terceira parte – referente à identificação de outros processos formais de avaliação, por qual tenha passado o programa em foco, que pudessem ser comparados, ou influenciarem na aplicação da avaliação pretendida;
- Processo de aquisição – referente ao relacionamento existente entre o processo de avaliação e o processo de aquisição.

A exclusão desses aspectos se deu pela irrelevância que apresentam para o objetivo deste estudo, uma vez que nenhum deles irá afetar, na prática, o desempenho das ferramentas, considerando que a avaliação destina-se a analisar os seus efeitos em um contexto específico de uso.

Com relação à gestão do processo de avaliação, encontram-se em ABNT (2003b) as prescrições referentes a esse tópico, conforme apresentado anteriormente no quadro 4. Observa-se que essa parte da norma está intimamente atrelada às etapas descritas no quadro 6, evidenciando-se a necessidade de documentar cada uma delas e atribuir responsabilidades aos participantes do processo, seja na condução ou no apoio; contudo, observa-se também que o viés das abordagens visa incentivar o aperfeiçoamento da metodologia de avaliação para indivíduos ou organizações que pretendem se profissionalizar neste tipo de atividade, uma vez que, apesar dos métodos apresentados possuírem caráter genérico, podendo ser seguidos, ou adaptados, de acordo com o objetivo proposto, resultam numa documentação extensa e, em muitos aspectos, redundante, se aplicada na íntegra ao formato de um trabalho como este.

Dessa forma, o Plano de Avaliação desenvolvido, no âmbito desse estudo, foi o resultado do detalhamento dos itens relacionados com a etapa de Projetação da Avaliação, constante do quadro 6, apresentado anteriormente.

Uma vez elaborado o Plano de Avaliação, partiu-se para a etapa de execução da avaliação que, de acordo com as prescrições constantes em ABNT (2004), deve ser devidamente documentada através da formulação do Módulo de Avaliação. O objetivo desse documento é garantir que o processo seja repetível, uma vez que nele estão contidos registros acerca do método, da técnica, das métricas aplicadas e dos resultados alcançados para cada uma das características de qualidade consideradas no procedimento avaliatório. Apesar da norma sistematizar um formato para a documentação do módulo de avaliação, os exemplos encontrados nos anexos da NBR ISO/IEC 14598-6 são de caráter genérico e declaradamente informativo, aplicáveis aos mais diversos contextos, podendo, assim como no plano de avaliação, serem seguidos ou adaptados de acordo com o objetivo proposto.

Considerando a natureza deste trabalho, a própria especificação das etapas do processo de avaliação, desenvolvida a seguir, somado aos relatos de sua elaboração, constituíram elementos de registro do método, da técnica e das métricas aplicadas, como forma de se guardar conformidade com o que está prescrito na norma.

Como resultado da execução da avaliação, foram desenvolvidos tópicos acerca da abrangência de deficiências específicas, limitações, condições especiais de uso dos *softwares*, bem como foram criadas fichas de avaliação para cada uma das ferramentas testadas a fim de que fossem registrados os resultados das medições, relacionados a cada uma das características de qualidade consideradas. Assim, criou-se as condições necessárias para que a avaliação possa ser revista e repetida, em condições similares à de sua elaboração, cumprindo a finalidade do módulo.

Uma vez definida a metodologia de avaliação dos *softwares* livres gráficos para formulação do experimento prático, explicado no capítulo 4, partiu-se para a escolha dos programas com base nessa mesma metodologia, considerando que a seleção foi feita objetivando chegar a um conjunto específico de ferramentas potencialmente capazes de atender às características de qualidade em uso necessárias para a aplicação em arquitetura.

### 3.6 Especificação dos Requisitos Para Escolha das Ferramentas

Para realização da avaliação foram utilizadas as seguintes categorias de *softwares* gráficos livres:

- *Software* CAD 2D, genérico, para elaboração de plantas técnicas de arquitetura;
- *Software* de Editoração Vetorial (Editor Gráfico de Vetores) para elaboração de peças ilustrativas do projeto;
- *Software* Modelador Geométrico Tridimensional com ferramenta de *Rendering* para elaboração de um modelo geométrico tridimensional e geração de imagens desse modelo.

Observe-se que as categorias de *softwares* aqui especificadas, inserem-se no conjunto de tecnologias computacionais aplicadas ao projeto que foram objeto de análise no capítulo 2 desse estudo.

As pesquisas desenvolvidas por Reis (2003, p.55) e Feller et al (2005, p. xix) fazem referência à página *web* Sourceforge (c2011), repositório de código fonte de projetos de desenvolvimento de *software* livre, como uma das principais fontes de consulta e acesso a esses programas. Uma visita a essa página *web* revelou que atualmente esse repositório hospeda 296.226 projetos de ferramentas livres para as mais diversas finalidades, sendo que desses 6.613 se enquadram na categoria *Graphic* das quais 3.426 são distribuídas sob os termos da licença GNU/GPL (SOURCEFORGE, c2011), que garante as liberdades que caracterizam o *software* livre considerado nesse trabalho.

Tratando-se de uma fonte referenciada na bibliografia consultada, e considerando o volume de projetos de *software* livre que hospeda, foi aqui também tomada como referência para busca e seleção dos programas gráficos a serem avaliados. Dentro desse universo, por meio do sistema de busca disponível no sourceforge.net, foram utilizadas palavras chaves, no idioma inglês, que direcionaram a pesquisa, de forma mais específica, a cada uma das categorias de programas anteriormente determinadas para realização do experimento, tais como: 2D CAD, 3D Modeler, 3D suit, Vector Graphics Editor etc., uma vez que o termo *Graphic* remete a um



universo muito amplo de programas, aplicáveis a uma grande diversidade de áreas do conhecimento. Com o resultado da pesquisa, foram consideradas a incidência dos seguintes fatores, como filtro, para uma primeira seleção:

8. O texto que descreve sumariamente a aplicação da ferramenta, disponível na própria página *web* sourceforge.net, observando se o mesmo oferece algum nível de compatibilidade com a aplicação necessária à realização do experimento ;
9. Estatística de *downloads*, dando-se preferência às ferramentas que angariam maior número de interessados;
10. Existência de uma página *web* dedicada à divulgação de informações sobre a ferramenta, suas potencialidades e o suporte em torno de seu uso.

Depois de passar por esses filtros, a escolha dos programas gráficos livres que compunham o "pacote" avaliado, foi definida com base nas características de qualidade, referenciadas na ABNT (2003a), constantes do quadro 7.

**Quadro 7** – Modelo de qualidade para seleção dos programas utilizados no experimento.

MODELO DE QUALIDADE EXTERNA DO PRODUTO DE <i>SOFTWARE</i>		
CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO
<b>Portabilidade</b>	Capacidade de ser instalado	Capacidade de ser instalado em um ambiente especificado.
<b>Funcionalidade</b>	Adequação	Capacidade de prover um conjunto apropriado de funções para realização das tarefas e objetivos.
<b>Usabilidade</b>	Apreensibilidade	Capacidade de possibilitar ao usuário aprender sua aplicação.

Fonte: Elaborado pelo autor com base em ABNT (2003a).

Os procedimentos para verificação do potencial atendimento, das ferramentas, às características de qualidade apresentadas no quadro 7 foram os seguintes:

- Os programas foram primeiramente submetidos à verificação do requisito de **Portabilidade**. O atendimento à subcaracterística de Capacidade de Instalação foi

considerado quando o programa apresentou condições de instalação em máquinas dotadas dos seguintes sistemas operacionais: Windows, MOSX e Linux. No tocante a esse requisito, observa-se que, dada a área do conhecimento a que se dedica esse estudo, Arquitetura e Urbanismo, não foram selecionados programas cujos arquivos de instalação requerem conhecimentos especializados de Ciência da Computação, por parte do avaliador e de usuários comuns, a exemplo dos arquivos binários que dependem da execução de *script*.

Da mesma forma, o experimento prático não foi aplicado em ambientes equipados com computadores que operam sobre a plataforma Linux, uma vez que a utilização desses equipamentos ainda não atingiu um grau significativo de difusão em instituições e meios corporativos de ensino e produção de projetos arquitetônicos, requerendo conhecimentos igualmente especializados para instalação dos programas. Contudo, a inclusão dessa plataforma no rol dos Sistemas Operativos que conferem a portabilidade requisitada dos programas trata-se de uma questão ética e condizente com o tema desse estudo.

- Os programas que atenderem à característica de portabilidade da etapa anterior foram submetidos à verificação dos requisitos de **Funcionalidade**, através de consulta à documentação disponibilizada na página *web* do programa. As funções que indicam o potencial da ferramenta atender à subcaracterística de Adequação, para cada um dos programas, estão descritas no quadro 8.
- Os programas que satisfizerem aos critérios das etapas anteriores foram submetidos à verificação dos requisitos de **Usabilidade**, pesquisando-se na página *web* do programa, a existência de tutoriais, manuais, livros e(ou) outras fontes de informação, de forma que fique comprovado que se oferece ao usuário material suficiente para aprender a operá-lo, ao nível básico, da aplicação das funções especificadas no quadro 8.

A execução dessas etapas de seleção resultou na escolha de apenas um *software* para cada categoria submetida ao processo de avaliação, ou seja; um *software* CAD 2D, um Editor Gráfico de Vetores e uma ferramenta de Modelagem Geométrica Tridimensional.

**Quadro 8** – Propriedades relacionadas com requisitos de Adequação

<i>SOFTWARE</i>	<b>CAD 2D</b>	<b>EDITORACÃO VETORIAL</b>	<b>MODELAGEM</b>
FUNÇÕES	<ul style="list-style-type: none"> <li>- Ferramentas para criar e modificar desenhos, utilizando um sistema cartesiano de coordenadas (x,y,z);</li> <li>- Oferecer opções de sistemas de unidades variado;</li> <li>- Dispor de recursos de edição que possibilitam:               <ol style="list-style-type: none"> <li>1. Deslocamentos (<i>Move</i>);</li> <li>2. Rotações (<i>Rotate</i>);</li> <li>3. Simetrias ou espelhamento (<i>Mirror</i>);</li> <li>4. Cópia simples (<i>Copy</i>) e cópias múltiplas (<i>Array</i>);</li> <li>5. Mudança de escala (<i>Scale</i>);</li> <li>6. Esticamento (<i>Stretch</i>), prolongamentos (<i>Extend</i>), encurtamentos (<i>Trim</i>) e rompimentos (<i>Break</i>);</li> <li>7. Chanfros (<i>Chamfer</i>) e concordâncias (<i>Fillet</i>);</li> </ol> </li> <li>- Possuir recursos de dimensionamento automático dos desenhos;</li> <li>- Recursos para mensurações e cálculo de distâncias, áreas e perímetros;</li> <li>- Permitir a divisão dos desenhos em camadas (<i>Layers</i>);</li> <li>- Possuir recurso de Bibliotecas de elementos gráficos;</li> <li>- Recursos para a inserção de Informações textuais;</li> <li>- Disponibilidade de <i>grids</i> e malhas</li> <li>- Recursos de precisão para seleção em pontos notáveis (<i>Snap</i>);</li> <li>- Recursos de visualização (zoom).</li> <li>- Recursos de importação e Exportação de Arquivos</li> </ul>	<ul style="list-style-type: none"> <li>- Ferramentas para criação e edição de formas geométricas, contornos e preenchimentos;</li> <li>- Oferecer opções de sistemas de unidades variados</li> <li>- Dispor de recursos de edição que possibilitam:               <ol style="list-style-type: none"> <li>1. Deslocamentos (<i>Move</i>);</li> <li>2. Rotações (<i>Rotate</i>);</li> <li>3. Simetrias ou espelhamento (<i>Mirror</i>);</li> <li>4. Cópia simples (<i>Copy</i>);</li> <li>5. Mudança de escala (<i>Scale</i>);</li> <li>6. Esticamento (<i>Stretch</i>);</li> </ol> </li> <li>- Possuir recursos para importação de imagens e criação de padrões de preenchimento;</li> <li>- Possuir recursos para edição de gradientes;</li> <li>- Permitir a divisão dos elementos em camadas (<i>Layers</i>);</li> <li>- Recursos para a inserção de Informações textuais;</li> <li>- Disponibilidade de <i>grids</i> e malhas</li> <li>- Recursos de precisão para seleção em pontos notáveis (<i>Snap</i>);</li> <li>- Recursos de visualização (zoom).</li> <li>- Recursos de importação e exportação de arquivos</li> </ul>	<ul style="list-style-type: none"> <li>- Ferramentas para criação e edição de formas geométricas tridimensionais, utilizando um sistema cartesiano de coordenadas (x,y,z);</li> <li>- Dispor de recursos de geração de modelos através do método CGS (<i>Construct Solid Geometry</i>);</li> <li>- Dispor de recursos para geração de modelos através do método B-Rep (<i>Boundary Representation</i>);</li> <li>- Capacidade para gerar modelos em <i>wireframe</i>, de superfície e sólidos;</li> <li>- Dispor de recursos de edição que possibilitam:               <ol style="list-style-type: none"> <li>1. Deslocamentos (<i>Move</i>);</li> <li>2. Rotações (<i>Rotate</i>);</li> <li>3. Simetrias ou espelhamento (<i>Mirror</i>);</li> <li>4. Cópia simples (<i>Copy</i>) e cópias múltiplas (<i>Array</i>);</li> <li>5. Mudança de escala (<i>Scale</i>);</li> <li>6. Extrusões;</li> </ol> </li> <li>- Permitir a divisão/agrupamento dos elementos em camadas (<i>Layers</i>)</li> <li>- Recursos de precisão para seleção em pontos notáveis (<i>Snap</i>);</li> <li>- Recursos de visualização (<i>zoom</i>)</li> <li>- Dispor de ferramenta de acabamento e <i>Rendering</i>;</li> <li>- Dispor de recursos de edição paramétrica</li> <li>- Recursos de importação e exportação de arquivo</li> </ul>

Fonte: Elaborado pelo autor.

### 3.7 Execução da Seleção dos Programas Livres Gráficos Avaliados.

O procedimento para pré-seleção dos programas, ocorrida na primeira fase de escolha, iniciou-se com a utilização de “palavras chaves” ou *topics*, para utilizar o termo encontrado na página *web* do sourceforge.net, aplicados ao seu sistema de busca. Considerando que esse estudo trata especificamente de programas livres distribuídos sob os termos da licença GNU/GPL, aplicou-se ao resultado da busca um primeiro filtro de seleção baseado nessa restrição, obtendo-se os resultados apresentados no quadro 9.

**Quadro 9** – Resultado da pesquisa de *softwares* livres gráficos na página *web* sourceforge.net

<i>TOPIC</i>	Nº de programas encontrados	Nº de programas com licença GNU/GPL
CAD – ( <i>Computer Aided Design</i> )	277	139
2D CAD	31	17
CAE – ( <i>Computer Aided Engineering</i> )	21	10
CADD – ( <i>Computer Aided Design and Drafting</i> )	1	0
CAAD – ( <i>Computer Aided Architectural Design</i> )	3	0
<i>Computer Aided Technologie</i>	136	67
<i>Vector Based</i>	101	51
<i>Graphic Editor</i>	384	221
<i>Vector Graphic Editor</i>	551	281
<i>Geometric Editor</i>	5	1
<i>3D Geometric Editor</i>	199	116
<i>3D Modeling</i>	510	261
3D CAD	56	34
3D <i>Suit</i>	52	27

Fonte: Quadro elaborado pelo autor, com base nos resultados da pesquisa realizada em Sourceforge (c2011).

As "palavras chaves" (*topics*) descritas no quadro 9, inseridas no sistema de busca da página *web* sourceforge.net, derivou dos termos atribuídos às tecnologias e ferramentas computacionais aplicadas ao projeto, vistas no capítulo 2 desse estudo. Durante esse procedimento, detectou-se que diversos programas figuravam em mais de uma das listas de resultados apresentadas com a utilização dessas “palavras chaves”, havendo programas que

por exemplo estavam presentes, simultaneamente, na listagem produzida com a aplicação dos termos *3D CAD*, *3D Modeling* e *3D Geometric Editor*, outros, concomitantemente nas listagem resultantes do uso dos termos *Vector Graphics Editor*, *Graphic Editor* e *CAD*, para citar alguns exemplos. Essas ocorrências, justificam a quantidade de programas atribuídos aos termos que apresentam números mais elevados, devido à própria abrangência do tópico, observando-se ainda que esses termos estão relacionados com tecnologias que não são de aplicação exclusiva ao campo do projeto de arquitetura.

Considerados apenas os programas distribuídos sob os termos da licença GNU/GPL, deu-se prosseguimento à aplicação dos filtros, estabelecidos na especificação dos requisitos para escolha das ferramentas, analisando-se os textos que resumem as principais características de cada uma delas, incluídos na listagem produzida pelo sistema de busca do site. Dessa forma, eliminou-se aquelas que entendeu-se não ter conformidade com os objetivos deste estudo, por mostrarem claramente, através da interpretação dos referidos textos, que destinavam-se a aplicações não empregadas na área do projeto de arquitetura.

Deve-se ressaltar, que um número expressivo de ferramentas eram dedicadas, de forma mais contundente, a atividades relacionadas com o desenvolvimento de jogos eletrônicos (*video games*) e de construção de interface gráfica para *softwares* com diversas finalidades.

Desse ponto em diante foi necessário organizar os quadros 10, 11 e 12, agrupando as ferramentas, que passaram pelos filtros já aplicados, e dispunham de página *web* dedicadas exclusivamente à divulgação de seus projeto de desenvolvimento, enquadrando-as de acordo com as categorias especificadas para realização do experimento prático, sejam elas: *CAD 2D*; *Editor Gráfico de Vetores* e *Modelador Geométrico Tridimensional*. Cabe observar que, essas categorias foram definidas anteriormente na especificação dos requisitos para escolha das ferramentas, assim denominadas para destacar a tecnologia predominante dos programas utilizados na elaboração das tarefas requeridas na execução do experimento prático, contudo, o processo aplicado nessa seleção mostrou que os *softwares* inclusos nessas categorias podem também estar relacionadas com outras tecnologias.

O processo de coleta das informações apresentadas nos quadros 10, 11 e 12, permitiu verificar que uma parcela dos *softwares* cadastrados no repositório *sourceforge.net*, não dispunham de informação acerca da estatística de *download*, ou mesmo dos próprios arquivos de instalação

dos programas, nesses casos recorreu-se às página *web* dedicadas a eles, onde constatou-se três situações:

9. Tratavam-se de projetos desativados ou sem manutenção. São os casos relacionados com número de *downloads* igual a zero,
10. Tratavam-se de projetos que dispunham dos arquivos apenas em repositório próprio.

No caso de ocorrência da segunda situação, encontrou-se situações em que a estatística constava no referido repositório, e uma única ocorrência em que não se dispunha de nenhuma estatística de *download*. Não obstante, houveram ainda situações em que encontrou-se os arquivos para instalação dos programas disponível para *download*, tanto na página *web* sourceforge.net, como nos sites a eles dedicados, contudo, nesses casos as estatísticas constavam apenas no primeiro.

**Quadro 10 – Pré seleção *software* CAD 2D**

SOFTWARE	PÁGINA WEB	DOWNLOADS		LICENÇA	SISTEMA OPERACIONAL
		Nº	PERÍODO		
Qcad Community	<a href="http://www.qcad.org">http://www.qcad.org</a>	15.878	Jul/10 – Jun/11	GNU/GPL	Linux; Windows
LibreCAD (Caduntun)	<a href="http://librecad.org">http://librecad.org</a>	13.529	Ago/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
Varkon	<a href="http://varkon.sourceforge.net/">http://varkon.sourceforge.net/</a>	5.516	Jul/10 – Jun/11	GNU/GPL	Linux; Windows
Blendercad	<a href="http://projects.blender.org/projects/blendercad/">http://projects.blender.org/projects/blendercad/</a>	3.998	Jul/10 – Jun/11	GNU/GPL	-
ThanCAD	<a href="http://thancad.sourceforge.net/">http://thancad.sourceforge.net/</a>	1.148	Jul/10 – Jun/11	GNU/GPL	Windows; OS independent
PythonCAD	<a href="http://pythoncad.org">http://pythoncad.org</a>	672	Jun/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
SagCAD	<a href="http://sagcad.sourceforge.jp">http://sagcad.sourceforge.jp</a>	385	Jul/10 – Jun/11	GNU/GPL	Linux; Solaris; BSD
Xdraft	<a href="http://xdraft.sourceforge.net/">http://xdraft.sourceforge.net/</a>	313	Jul/10 – Jun/11	GNU/GPL	Linux; Solaris; BSD
ConstructCAD	<a href="http://concad.sourceforge.net/">http://concad.sourceforge.net/</a>	0	Jul/10 – Jun/11	GNU/GPL	Linux; Windows
gCAD	<a href="http://www.gcad3d.org/">http://www.gcad3d.org/</a>	0	Jul/10 – Jun/11	GNU/GPL	Linux; Windows
Jcad	<a href="http://javacad.sourceforge.net">http://javacad.sourceforge.net</a>	0	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
oCADIS	<a href="http://www.nocrew.org/software/ocadis">http://www.nocrew.org/software/ocadis</a>	0	Jul/10 – Jun/11	GNU/GPL	Linux; Solaris

Fonte: Elaborado pelo autor.

Observando-se o quadro 10 verifica-se que a metodologia adotada para escolha dos programas na categoria CAD 2D levou inicialmente à pré-seleção de doze programas que satisfizeram aos critérios aplicados até aquele ponto. Com relação à ferramenta Blendercad, trata-se do projeto de um aplicativo (*plug-in*) criado para "rodar" sobre o Blender, *software* de modelagem tridimensional que será visto adiante, contudo não foi possível identificar, para

quais sistemas operativos esse aplicativo está habilitado.

**Quadro 11** – Pré seleção *software Vector Graphic Editor*

SOFTWARE	PÁGINA WEB	DOWNLOADS		LICENÇA	SISTEMA OPERACIONAL
		Nº	PERÍODO		
Inkscape	<a href="http://BrOffice.org/">http://BrOffice.org/</a>	3,925,865	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
Synfig	<a href="http://synfig.org/">http://synfig.org/</a>	133.688	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
Skencil	<a href="http://www.skencil.org/">http://www.skencil.org/</a>	6.128	Jul/10 – Jun/11	GNU/GPL	Linux; MOSX
sK1	<a href="http://sk1project.org/">http://sk1project.org/</a>	179	Jul/10 – Jun/11	GNU/GPL	Linux
Koffice/Karbon 14	<a href="http://www.koffice.org/karbon/">http://www.koffice.org/karbon/</a>	?	?	GNU/GPL	Linux; Windows; MOSX

Fonte: Elaborado pelo autor.

O quadro 11 demonstra que dentro do universo dos *softwares* livres registrados no sourceforge.net, depois de aplicados os filtros iniciais de seleção, restaram apenas cinco deles enquadrados na categoria *Vector Graphic Editor*, sendo que não se encontrou estatística de número de *downloads* do programa Karbon 14, nem na página *web* sourceforge.net, nem na página dedicada ao programa, que é um aplicativo para elaboração e editoração de desenhos e imagens vetoriais, integrante do Koffice, versão de suite de escritório distribuída junto com a interface gráfica KDE para Linux, mas também já existente em versões para Windows e MOSX (KOFFICE, 2011).

**Quadro 12** – Pré seleção *software CAD 3D*

SOFTWARE	PÁGINA WEB	DOWNLOADS		LICENÇA	SISTEMA OPERACIONAL
		Nº	PERÍODO		
Blender	<a href="http://www.blender.org/">http://www.blender.org/</a>	3.400.000	Média Anual (2009)	GNU/GPL	Linux; Windows; MOSX; BSD
K3D	<a href="http://www.k-3d.org/">http://www.k-3d.org/</a>	157.209	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX; BSD; Solaris
AutoQ3D	<a href="http://www.autoq3d.com/">http://www.autoq3d.com/</a>	23.992	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
DeleD Community Edition	<a href="http://www.delgine.com/">http://www.delgine.com/</a>	14.418	Jul/10 – Jun/11	GNU/GPL	Windows
AvoCADo-CAD	<a href="http://avocado-cad.sourceforge.net/">http://avocado-cad.sourceforge.net/</a>	10.848	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
Varkon	<a href="http://www.varkon.org/">http://www.varkon.org/</a>	5.620	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; BSD
LignumCad	<a href="http://lignumcad.sourceforge.net/doc/en/HTML/index.html">http://lignumcad.sourceforge.net/doc/en/HTML/index.html</a>	1.764	Jul/10 – Jun/11	GNU/GPL	Cross Platform
SagCAD	<a href="http://sagcad.sourceforge.jp">http://sagcad.sourceforge.jp</a>	385	Jul/10 – Jun/11	GNU/GPL	Linux; Solaris; BSD
gCAD	<a href="http://www.gcad3d.org/">http://www.gcad3d.org/</a>	0	Jul/10 – Jun/11	GNU/GPL	Linux; Windows

Fonte: Elaborado pelo autor.

Conforme pode ser observado no quadro 12, na primeira fase de triagem, apenas nove *softwares* foram pré-selecionados na categoria CAD 3D. Sendo que, se pode perceber, através das estatísticas de *download*, que o interesse por essas ferramentas é, em média, superior ao das outras categorias de programas pesquisadas. No caso do Blender, encontrou-se estatística atualizada acerca do número de acessos à página *web* dedicada ao programa que, no período de abril de 2010 a abril de 2011, registrou 14.354.707 visitantes, oriundos de 231 países (BLENDER FOUNDATION, 2011f), contudo, referente ao número de *downloads* verificou-se registro referente ao período de julho de 2004 a julho de 2005, quando se apurou 1.823.710 transferências dos arquivos de instalação do programa, e o último registro data do ano de 2009, apurado no período de abril de 2008 a abril de 2009, apresentado acima no quadro 12 (BLENDER FOUNDATION, 2011e). Com base nesses números considerou-se que o programa ocupa o topo do *ranking* de *downloads* entre as ferramentas 3D pré-selecionadas.

Depois de finalizada essa primeira fase do processo de seleção dos programas com potencial de aplicação para o experimento prático desse estudo, concluiu-se que apesar do sourceforge.net hospedar centenas de projetos de programas livres gráficos, existe um número muito elevado desses que não tiveram continuidade e estão estagnados. Em alguns casos, não chegou sequer a ser lançada uma versão experimental do programa ou, então, não foram dotados de uma quantidade suficiente de recursos para o uso prático. Constatou-se também que existe uma diversidade de licenças atribuídas a esses *softwares*, às quais Laurent (2004, p.4) diz ser conhecido popularmente como *Open Source Licensing*, a exemplo da MIT *Licensing*, BSD *Licensing* e Qt *Licensing*, contudo, o autor afirma também que a GNU/GPL é aquela que impõe de forma mais específica as limitações na forma em que os programas derivados de *softwares* de código aberto podem ser licenciados garantindo, de forma mais eficiente, as liberdades fundamentais que caracterizam o *software* livre (LAURENT, 2004, p.34).

Em uma segunda fase aplicou-se, ao elenco de ferramentas pré-selecionadas, os filtros definidos com base nas características de qualidade referenciadas na ABNT (2003a), constante do quadro 7, apresentado anteriormente. De acordo com os procedimentos estabelecidos, para aplicação desses filtros, eliminou-se primeiramente aquelas ferramentas que não guardavam conformidade com a característica de portabilidade, fazendo-se jus ao requisito referente aos Sistemas Operativos com os quais as ferramentas deviam ser compatíveis, sejam eles: Linux, Windows e MOSX. A aplicação desse critério reduziu



significativamente a quantidade de programas selecionados, a seguir apresentados nos quadros 13, 14 e 15.

**Quadro 13** – Programas CAD 2D selecionados.

SOFTWARE	PÁGINA WEB	DOWNLOADS		LICENÇA	SISTEMA OPERACIONAL
		Nº	PERÍODO		
LibreCAD (Caduntun)	<a href="http://librecad.org">http://librecad.org</a>	13.529	Ago/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
ThanCAD	<a href="http://thancad.sourceforge.net">http://thancad.sourceforge.net</a>	1.148	Jul/10 – Jun/11	GNU/GPL	Windows; OS Independent
PythonCAD	<a href="http://pythoncad.org">http://pythoncad.org</a>	672	Jun/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX

Fonte: Elaborado pelo autor.

Entre os doze programas CAD 2D pré-selecionados anteriormente, apenas os três constantes do quadro 13, guardam conformidade com o requisito que define a portabilidade especificada para os *softwares*. Entretanto, pesquisando-se os arquivos de instalação dessas ferramentas, bem como a documentação disponível em suas página *web*, observou-se que o Than CAD e o Python CAD disponibilizam arquivos na forma binária, o que confere ao primeiro deles a característica de OS *Independent*, ou seja, seus arquivos podem ser compilados para qualquer sistema operacional, por qualquer um que saiba manipular esse tipo de arquivo, sendo que a versão para Windows já é disponibilizada em formato executável. Assim restou o Caduntun como único programa apto para prosseguimento nas investigações.

Aprofundada, então, a investigação na página *web* do programa, constatou-se que o projeto denominado LibreCAD deu origem ao Caduntun a partir da versão *Community* (livre) do *software* QCAD, cujo desenvolvimento, nos moldes de produção do *software* livre, foi interrompido pela empresa Ribbonsoft que o mantinha e passou a desenvolver e comercializar uma versão proprietária do *software* (RIBBONSOFT, 2011b), embora, talvez por força dos termos da licença GNU/GPL, continue disponibilizando o código fonte da última versão *freeware* do programa. Reportando-se ao quadro 10, elaborado anteriormente, observa-se que, inclusive, essa ferramenta lidera o *ranking* de *downloads* na categoria CAD 2D da página *web* sourceforge.net, entretanto, só está disponível em versões para Linux e Windows. Nesse contexto, o projeto LibreCAD, aproveitou o código fonte do QCAD e retomou seu desenvolvimento a partir dos recursos e da interface existentes, modificando apenas sua denominação para Caduntun. Relativo à característica de Usabilidade, o projeto LibreCAD

mantém páginas *web* dedicadas ao auxílio de usuários e desenvolvedores do programa (LIBRECAD, 2011), contudo, a melhor fonte de suporte encontrada para o aprendizado de suas funcionalidades foi o manual do usuário do QCAD, (RIBBONSOFT, 2011a) que ensina como utilizar todos os comandos do programa. Diante da facilidade de instalação e do contexto favorável à aplicação prática do programa, o Caduntun foi selecionado e definido para execução do experimento prático deste estudo.

**Quadro 14** – Programas *Vector Graphic Editor* selecionados.

SOFTWARE	PÁGINA WEB	DOWNLOADS		LICENÇA	SISTEMA OPERACIONAL
		Nº	PERÍODO		
Inkscape	<a href="http://inkscape.org/">http://inkscape.org/</a>	3.925,865	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
Synfig	<a href="http://synfig.org/">http://synfig.org/</a>	133.688	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
Koffice/Karbon 14	<a href="http://www.koffice.org/karbon/">http://www.koffice.org/karbon/</a>	?	?	GNU/GPL	Linux; Windows; MOSX

Fonte: Elaborado pelo autor.

Aplicando-se o filtro de portabilidade, dois dos cinco programas pré selecionados na categoria *Vector Graphic Editor* foram descartados, sendo que dos três restantes, apresentados no quadro acima, constatou-se que o Koffice/Karbon 14, além de não dispor de estatística de *download*, depende da execução de *script* para ser instalado no sistema MOSX, não satisfazendo, portanto, ao requisito de facilidade de instalação anteriormente especificado.

Nesse contexto, deu-se prosseguimento à aplicação dos critérios de seleção investigando-se o conteúdo das páginas *web* dedicadas aos projetos de desenvolvimento do Synfig e do Inkscape. Observou-se que ambos os programas possuem as características de funcionalidade requisitadas dispondo dos recursos necessários para a elaboração da ilustração produzida durante a execução do experimento prático desse estudo. Contudo, observou-se que o Synfig é um programa cujo propósito de desenvolvimento foi a criação de figuras 2D animadas, para produção de filmes (SYNFIG STUDIO, 2011), enquanto o projeto do Inkscape buscou o desenvolvimento de uma ferramenta livre para produção de ilustrações, seguindo o padrão de programas proprietários similares como o Adobe Illustrator e o Corel Draw (INKSCAPE.ORG, 2011). Constatou-se que ambos os programas dispõem de página *web* em idioma português, mantidas por comunidade de brasileiros, no entanto, no caso do Synfig trata-se de uma comunidade de usuários, empenhada na divulgação do programa (SYNFIG BR, 2011). No

caso do Inkscape, essa comunidade é formada por usuários, programadores e colaboradores ativos que participam do projeto de desenvolvimento do *software* gerando, portanto, um conteúdo mais diversificado para a página *web* do programa, disponibilizando extenso material relacionado com o suporte ao usuário formado por tutorias, artigos, bibliografia, *chat* e lista de discussão, entre outros, para citar os principais meios oferecidos (INKSCAPE BRASIL, 2011).

Com base nos atributos funcionais, na disponibilidade de material de suporte ao aprendizado, publicado em idioma português, na popularidade do programa, e na facilidade de instalação, o Inkscape foi selecionado entre os *softwares* livres gráficos da categoria *Vector Graphic Editor* para aplicação no experimento prático deste trabalho.

**Quadro 15** – Programas CAD 3D selecionados.

SOFTWARE	PÁGINA WEB	DOWNLOADS		LICENÇA	SISTEMA OPERACIONAL
		Nº	PERÍODO		
Blender	<a href="http://www.blender.org">http://www.blender.org</a>	3.400.000	Média anual (2009)	GNU/GPL	Linux; Windows; MOSX; BSD
K3D	<a href="http://www.k-3d.org/">http://www.k-3d.org/</a>	157.209	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX; BSD; Solaris
AutoQ3D	<a href="http://www.autoq3d.com/">http://www.autoq3d.com/</a>	23.992	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
AvoCADO-CAD	<a href="http://avocado-cad.sourceforge.net/">http://avocado-cad.sourceforge.net/</a>	10.848	Jul/10 – Jun/11	GNU/GPL	Linux; Windows; MOSX
LignumCad	<a href="http://lignumcad.sourceforge.net/doc/en/HTML/index">http://lignumcad.sourceforge.net/doc/en/HTML/index</a>	1.764	Jul/10 – Jun/11	GNU/GPL	Cross Platform

Fonte: Elaborado pelo autor.

De acordo com o quadro 15, cinco programas na categoria CAD 3D estão aptos a “rodar” sobre os sistemas operativos especificados, contudo, os arquivos de instalação dos *softwares* AutoQ3D, AvoCADO-CAD e LignumCAD, são disponibilizados apenas na versão binária. Além disso, como pode ser observado no quadro acima, o interesse pelo K3D e pelo Blender é muito superior ao despertado pelos outros programas. Assim, o processo de definição do *software* utilizado no experimento prático seguiu com esses dois últimos programas.

A pesquisa realizada na página *web* do K3D mostrou que o programa é dotado dos atributos que confere as características funcionais especificadas para realização do experimento prático desse estudo, dispões de manual exclusivamente no idioma inglês com informações gerais

acerca do uso de suas principais ferramentas, dispõe ainda de alguns vídeo tutoriais e uma sessão de FAQ (*Frequent Asked Questions*) que reúne perguntas e respostas mais frequentes derivadas de sua *Mailing List*, que é uma página *web* onde usuários cadastrados podem enviar mensagens, no idioma inglês, para a equipe de desenvolvedores do programa a fim de esclarecer dúvidas sobre sua utilização (K3D.ORG, 2011). Avançando-se na exploração da página *web* do K3D e nos Blogs a que faz referência, encontrou-se apenas um único exemplo de uso do programa relacionado com o campo da arquitetura, em uma aplicação para fins de estudo de insulejamento, produzida com auxílio de um aplicativo de renderização (MATTHEWS, 2011).

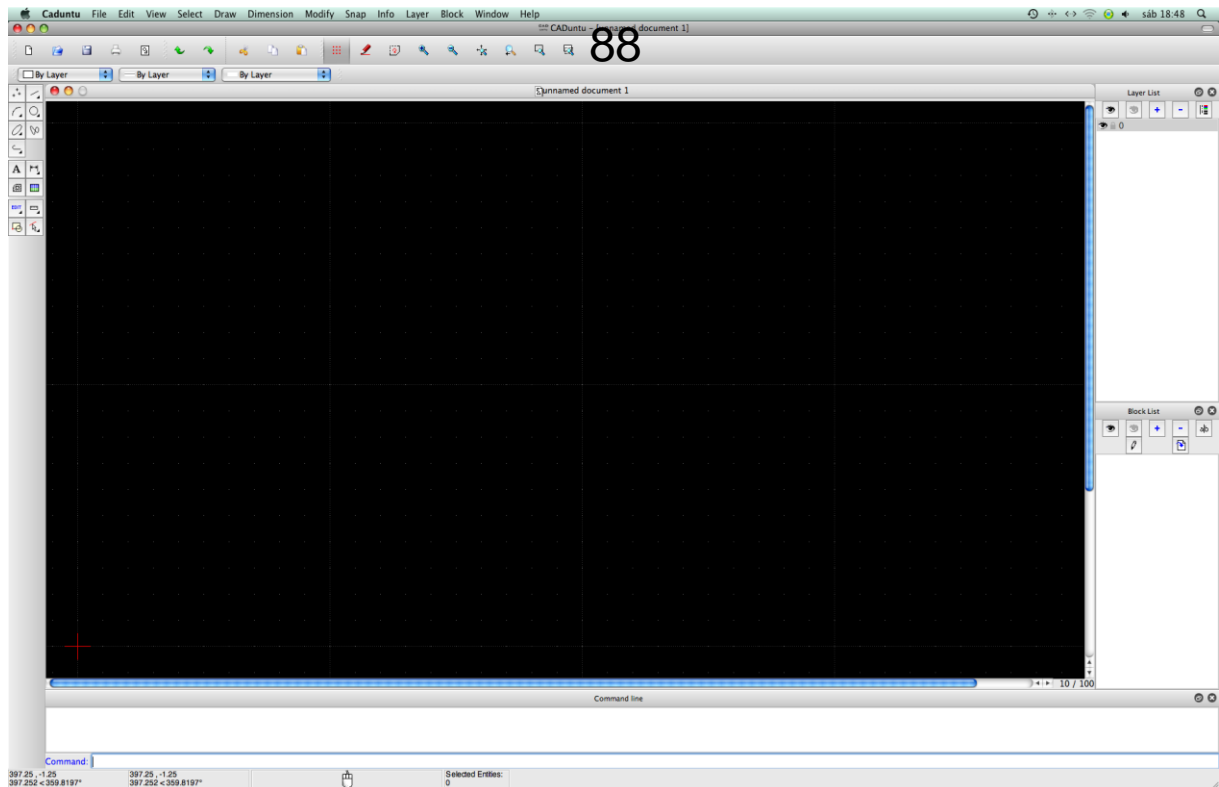
Com relação a investigação realizada na página *web* do Blender, observou-se que a estrutura de desenvolvimento e manutenção do programa é suportada pela *Blender Foundation*, organização sem fins lucrativos vinculada ao *Blender Institute*, com sede em Amsterdã, na Holanda (BLENDER FOUNDATION, 2011b). Com base nessa estrutura organizacional, o projeto Blender mantém uma extensa comunidade de usuários e desenvolvedores de forma que, atualmente, existem páginas *web* dedicadas ao programa em 18 diferentes idiomas, inclusive o português do Brasil, onde se pode encontrar tutoriais, bibliografia, fóruns de discussão, agenda de eventos e encontros, geograficamente localizados, em torno do uso do *software* (BLENDER FOUNDATION, 2011c). O programa reúne um conjunto de ferramenta para criação de conteúdo 3D, voltado principalmente para a produção de animações de curta e longa metragem, e jogos eletrônicos (*video game*), contudo verifica-se na seção *Art Gallery*, que não são raros os exemplos de imagens geradas a partir de modelos tridimensionais de ambientes internos, externos, edifícios e contextos urbanos (BLENDER FOUNDATION, 2011a) elaborado com o emprego do programa, uma vez que suas funcionalidades e recursos disponíveis, atendem de forma satisfatória a esses propósitos. Com base nesse diagnóstico, face à grande popularidade do Blender e ao extenso material disponível para seu aprendizado ele foi selecionado para aplicação no experimento prático desse estudo.

### **3.8 Programas Livres Gráficos Selecionados**

#### **Caduntun**

O Caduntun é um aplicativo de desenho auxiliado por computador com o qual é possível criar desenhos técnicos. Sua Interface gráfica é bastante intuitiva, similar a dos programas CAD

mais conhecidos (figura 11).



**Figura 11** – Interface gráfica do Caduntun.  
Fonte: Elaborado pelo autor.

O *software* possui um aplicativo de configurações do sistema que permite alterar a aparência de sua interface gráfica no que se refere à linguagem utilizada, aceitando até vinte e seis diferentes idiomas, inclusive o português do Brasil. Esse aplicativo possibilita também modificar as dimensões do *grid*<sup>64</sup>; que serve como referência do espaço da área de trabalho, cores aplicadas à "janela" de desenho (*background*), bem como o sistema de unidades de medida, que permite até vinte padrões diferenciados de dimensionamento, entre eles o sistema métrico e de polegadas, utilizados na arquitetura. Essas possibilidades de configuração são importantes para a adaptação do programa às preferências e necessidades do usuário.

O Caduntun possui recursos adequadas para criação e edição de desenhos e textos, comumente encontradas em ferramentas CAD 2D, cobrindo todas os atributos funcionais especificados anteriormente no quadro 8. Os desenhos, construídos a partir da edição de

<sup>64</sup> Malha de pontos mostrada na área de desenho.

primitivas geométricas; ponto, linhas, arcos, polígonos, etc... podem ser realizados com alto nível de precisão, organizados em camadas (*layers*), cujo sistema de gerenciamento consiste na atribuição de um nome, para cada camada, e cores aos elementos do desenho, relacionadas com a espessura do traço quando o arquivo gráfico der saída por meio de impressão. Conforme visto no capítulo 2, a utilização de camadas se tornou um padrão de gerenciamento de desenhos adotado em sistemas CAD. A saída gráfica dos arquivos do Caduntun se dá por meio do monitor, auxiliado pelo sistema de visualização do programa, dotado de recursos de *zoom*, comumente empregados em ferramentas dessa natureza, que amplia e reduz o desenho, infinitamente, ou por meio da impressão, na escala desejada pelo usuário, admitindo trinta diferentes formatos de papel, entre eles os da série A definidos pela ABNT (1994), além de formatos customizados. Dessa forma, o tamanho da folha de desenho, no dispositivo de saída, depende mais do *hardware* utilizado do que do programa.

O formato de arquivo nativo do programa é o DXF (*Drawing Exchange Format*) que é um formato padronizado para intercâmbio de arquivos de CAD, aceito pela grande maioria de programas dessa categoria, como o AutoCAD da Autodesk e por alguns programas de ilustração como o Corel Draw. A exportação de arquivos no programa se limita a formatos de dados de imagem png<sup>65</sup>, bastante difundido e aceito por uma vasta gama de programas e os formatos ppm<sup>66</sup>, xbm e xpm<sup>67</sup>, de aceitação mais restrita. Apesar da existência de um comando de importação de arquivos estar presente no menu de arquivos do programa, suas funcionalidades ainda estão em desenvolvimento. A página *web* dedicada ao projeto de desenvolvimento do Caduntun disponibiliza uma rede de suporte ao usuário baseada em lista de mensagens e fóruns de discussão (LIBRACAD, 2011). Não foi encontrado manual de orientação ao usuário elaborado no âmbito do projeto de desenvolvimento do programa, contudo, sendo esse uma continuação do desenvolvimento da versão livre do Qcad da Ribbonsoft (2011a; 2011b) o manual do usuário deste último programa, disponível na internet, se aplica ao aprendizado do Caduntun.

## **Inkscape**

O Inkscape é um programa que implementa o padrão SVG (*Scalar Vector Graphics*) que

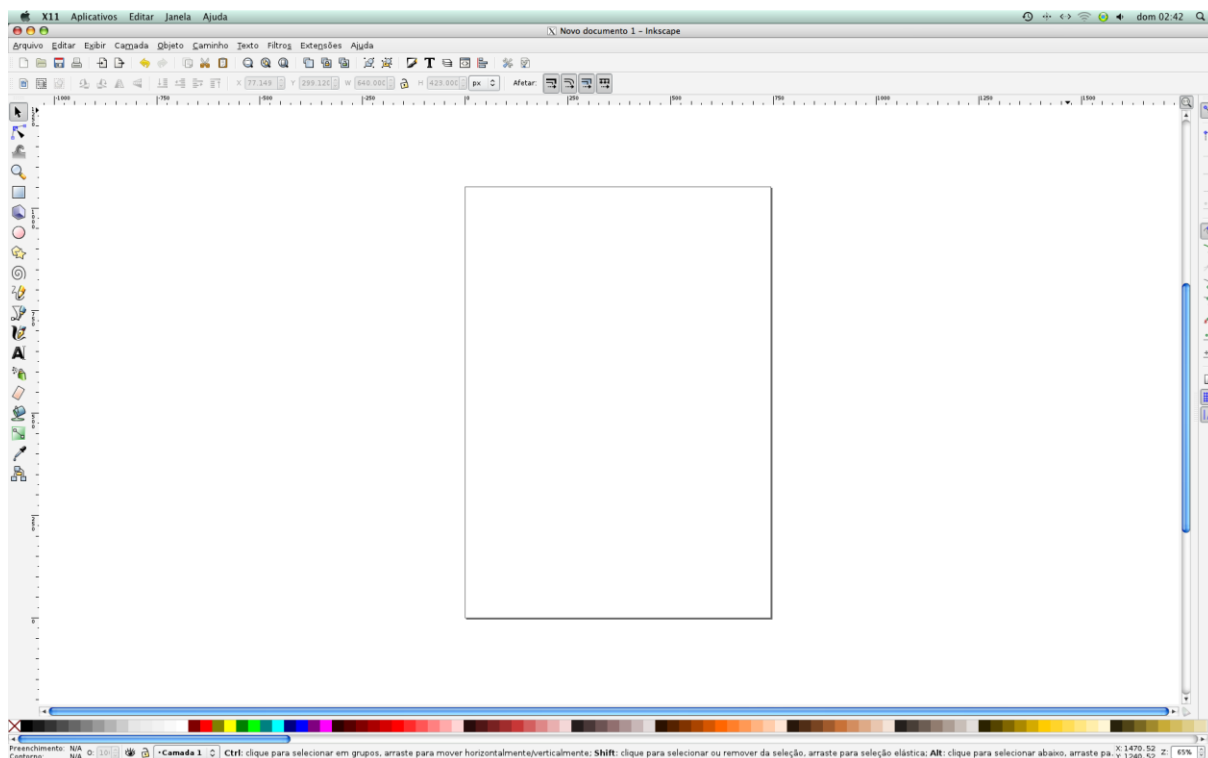
---

<sup>65</sup> *Portable Network Graphic* (PNG):

<sup>66</sup> *Portable pixmap* (PPM)

<sup>67</sup> X bitmap (XBM) é um formato de dados de imagem em preto e branco utilizado pelo sistema X Window e o X pixmap é uma evolução do primeiro que aceita cores.

suporta a construção de desenhos de formas complexas em duas dimensões. A figura 12 apresenta a interface gráfica do programa.



**Figura 12** – Interface gráfica do Inkscape.  
Fonte: Elaborado pelo autor.

Com base na figura 11, observa-se que a aparência do Inkscape é muito próxima da do Corel Draw, *software* da Corel Corporation, tanto quanto os recursos que dispõe. Com ele é possível desenhar formas geométricas básicas e modificá-las, através da manipulação dos "nós"; pontos que controlam a forma das curvas que os interliga, para construir formas mais complexas. Trabalha-se no programa com base na criação de contornos (vetores) e no preenchimento das áreas por eles formadas, com cores sólidas ou gradiente de cores, possibilitando uma variada gama de efeitos que, dependendo da habilidade e do domínio que o usuário tenha sobre o programa, pode atingir matizes realísticas. Na arquitetura esse recurso é muito empregado para a produção de apresentações e de material gráfico publicitário, explorado principalmente pelo mercado imobiliário, para auxiliar na compreensão do projeto por parte de leigos.

O Inkscape possui sistema de configuração das propriedades do desenho que permite

controlar o tamanho do formato do papel, exibido na tela, correspondente a uma folha de desenho no tamanho real, de forma que o usuário pode trabalhar diretamente sobre essa “folha de papel” visualizando como os gráficos estarão nela dispostos após a impressão. Uma malha de linhas pode ser aplicada na área de trabalho, associada a ferramentas de precisão (*snap*) para melhor controle das dimensões e da escala do desenho, que também pode ser atribuída através do fornecimento de parâmetros em campos específicos relacionados com a geometria dos objetos criados. Similarmente aos editores de desenho dos programas CAD, o inkscape possui ferramentas de gerenciamento de camadas que lhe confere muitas possibilidades de sobreposição dos desenhos, aliada a poderosas ferramentas de gerenciamento de cores, gradientes e transparência que são de grande utilidade na composição dos mais diversos tipos de efeitos de volume, luz e sombra.

O programa adota o padrão SVG de arquivo que é um formato de dado que pode ser aberto por qualquer *web browser* (navegador internet) sendo capaz de importar uma gama muito variada de formato de dados, principalmente os de imagem e vetoriais, mas, exporta apenas o formato Bitmap (imagem raster) no padrão “png”, não obstante, salva vinte e dois diferentes padrões de arquivo, entre eles o “pdf” em meio a outros menos difundidos.

Uma rede internacional de desenvolvedores, com representação expressiva no Brasil se dedica ao desenvolvimento contínuo do programa. Na página *web* da comunidade brasileira do Inkscape encontram-se diversas informações a seu respeito, desde o histórico de seu desenvolvimento até exemplos de trabalhos realizados com o uso do programa, tutoriais<sup>68</sup> e cursos *online* (INKSCAPE BRASIL, 2011), oferecendo boas condições para o aprendizado auto didata.

## **Blender**

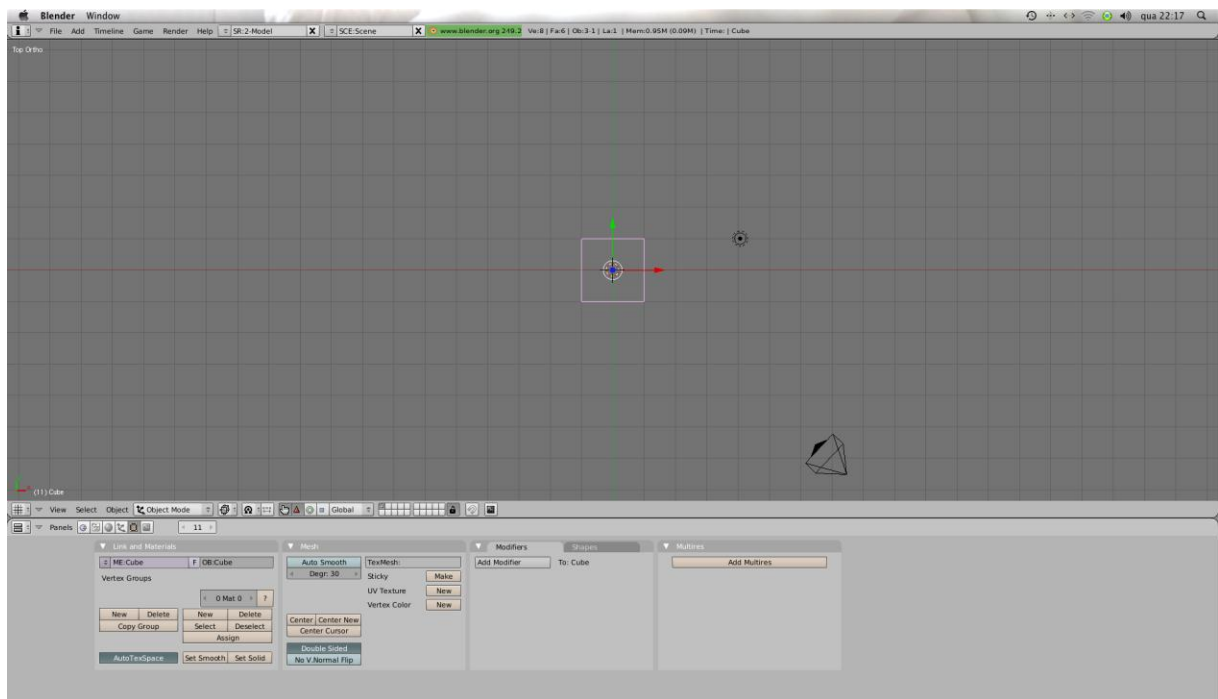
Apesar de ser direcionado para artistas gráficos, o Blender é apropriado para a modelagem tridimensional em arquitetura, permitindo gerar imagens estáticas e dinâmicas do modelo, inclusive com recursos de interação com a cena criada, como *walk-through*, e recursos mais sofisticados utilizados em Realidade Virtual, como detecção de colisão, modelagem e

---

<sup>68</sup> Tutorial é uma ferramenta de ensino/aprendizagem, podendo ser tanto um programa de computador quanto um texto, contendo ou não imagens, que auxilia o processo de aprendizado exibindo passo a passo o funcionamento de algo.



animação procedurais. A interface gráfica do programa é mostrada na figura 13.



**Figura 13** – Interface gráfica do Blender.

Fonte: Elaborado pelo autor.

A interface do programa é formada por “janelas” que podem ser organizadas de formas variadas, serem vistas ou ocultadas de acordo com a ferramenta que esteja sendo utilizada e com a própria preferência do usuário. Na figura acima, visualiza-se a tela de abertura do programa que dispõe de um menu na barra superior, onde encontram-se os principais comandos de modelagem, animação e renderização. Logo abaixo dessa barra está a *3D view window* (janela de visualização 3D), através da qual é possível visualizar e manipular as malhas, como são chamados os objetos tridimensionais.

O quadrado cor de rosa visualizado no centro da *3D view window* (figura 13) é a representação de um cubo em uma vista superior, sobre o qual está o manipulador de movimento, representando os três eixos que compõem o sistema de coordenadas espacial. Observa-se também nessa janela a presença de um ponto preto que representa um elemento de iluminação, o qual pode ser um sol ou uma lâmpada, e uma câmera. A correta manipulação desses elementos, através das ferramentas de edição e aplicação de materiais, encontradas nos painéis de botões, dispostos na janela inferior da interface gráfica, permitirá a construção dos modelos e a criação de cenas que, depois de renderizadas, geram imagens com alto grau de

realismo, como a apresentada na figura 14, extraída dos arquivos de imagem da página *web* dedicada ao programa (BLENDER FOUNDATION, 2011a).



**Figura 14** – Imagem de modelo construído com o Blender.

Fonte: (BLENDER FOUNDATION, 2011a).

Nesse contexto, observa-se que o programa possui ferramentas avançadas de modelagem, animação e *rendering* com vários exemplos de uso profissional, inclusive na área de arquitetura, como dito anteriormente, e apresentado na figura acima, mas, principalmente, é empregado na indústria cinematográfica e de produção de jogos eletrônicos, onde se originou (BLENDER FOUNDATION, 2011b). Devido a esse alcance profissional, o programa possui recursos de importação e exportação de diversos formatos de arquivos gráficos, de vídeo e de imagem, entre eles; dxf, svg e 3ds, esse último formato nativo do 3ds Max da Autodesk, sendo que o seu formato de arquivo nativo é o “.blend”.

Observa-se na página *web* do Blender que outros projetos de *software* livre estão relacionados com o uso e aperfeiçoamento do programa, que, em consequência de sua arquitetura modular, gradativamente, absorve novas funcionalidade oriundas desse processo paralelo de desenvolvimento coordenado pela Blender Foundation.

Diante do exposto, conclui-se que o processo de seleção adotado para escolha dos programas definiu a formação de uma “pacote” de *softwares* livres gráficos compatíveis com os atributos

de adequação funcional requisitados no âmbito da especificação das características de qualidade em uso, que fez parte do processo de avaliação dos mesmos, deixando-se aqui um panorama generalizado acerca das suas principais funcionalidades, com foco no contexto do experimento prático descrito no capítulo seguinte, ao final do qual obteve-se os elementos necessários para uma análise mais cuidadosa acerca do potencial de emprego dessas ferramenta na prática de projetos de arquitetura.

## **4. UMA EXPERIÊNCIA DE AVALIAÇÃO EM USO**

Este capítulo, trata da execução do experimento prático de uso de um "pacote" de programas livres gráficos, composto de três ferramentas selecionadas anteriormente, sejam elas: Caduntun, Inkscape e Blender, utilizadas no desenvolvimento de etapas específicas de um projeto de arquitetura. Durante o experimento foram coletados dados de ordem quantitativa e qualitativa. Os primeiros obtidos através da aplicação dos métodos e técnicas de avaliação de *software* prescritos pela ABNT (2003c), sintetizados no quadro 6 elaborado no título anterior, visando quantificar variáveis ou características de qualidade em uso das ferramentas, referenciadas no modelo definido na ABNT (2003a). Concomitante a esse processo, foi feito o registro das ocorrências com base na observação do grupo de usuários, suas reações individuais e impressões acerca da apreensibilidade, usabilidade e resultados alcançados com o uso dos programas, obtendo-se, assim, dados de ordem qualitativa, considerado o ponto de vista desse grupo específico de usuários. De acordo com Neves (1996, p. 1) [...] *um estudo de pesquisa qualitativa supõe um "corte" temporal-espacial* [...] que define o contexto de desenvolvimento do trabalho, e neste sentido a especificação detalhada do contexto de uso, requerida pelo método adotado para obtenção dos dados quantitativos, serviu também para contextualizar o ambiente, os agentes envolvidos e as circunstâncias mapeadas pela abordagem qualitativa. Por fim, buscou-se fazer uma análise interpretativa relacionando dados quantitativos e qualitativos, ao que Jick (1979) apud Neves (1996, p. 2) chama de "triangulação", resultando na avaliação propriamente dita que subsidiou a formulação de sentenças acerca do emprego desses programas na prática da arquitetura, bem como de recomendações para trabalhos futuros, dentro dessa temática, colocados no capítulo conclusivo.

#### **4.1 Especificação dos Requisitos de Avaliação**

O modelo proposto para execução do experimento prático é um Módulo Policial, similar ao padrão encontrado na cidade de Salvador, originalmente elaborado pela DESAL (Companhia de Desenvolvimento Urbano de Salvador). O experimento foi realizado em três ambientes distintos, tendo como público alvo arquitetos e estudantes de Arquitetura, que neles atuam profissionalmente ou na condição de estagiários, desempenhando atividade de desenvolvimento de projetos arquitetônicos e de urbanização com auxílio de ferramentas computacionais. O experimento consistiu no emprego exclusivo dos referidos programas livres gráficos para obtenção dos produtos indicados a seguir:

### 1 - Caduntun:

Representação gráfica da planta baixa do Módulo, contendo seus elementos construtivos, equipamentos fixos e mobiliário, com indicação de medidas, níveis, áreas, denominação dos compartimentos e indicação de secção, seguindo os padrões e as simbologias prescritas pela ABNT (1994). Ver APÊNDICE D;

Representação gráfica de um corte transversal da edificação, contendo seus elementos construtivos, seccionados e vistos, com indicação de medidas e níveis, seguindo os padrões e as simbologias prescritas pela ABNT (1994). Ver APÊNDICE D;

Representação gráfica das fachadas frontal e lateral da edificação, seguindo os padrões e as simbologias prescritas pela ABNT (1994). Ver APÊNDICE D.

### 2 – Inkscape:

Ilustração da Planta Baixa e das Fachadas, frontal e lateral, referenciada em padrões comumente adotados pelo mercado imobiliário de Salvador. Ver APÊNDICE E;

### 3 - Blender:

Modelo geométrico tridimensional da edificação, contemplando a representação dos elementos arquitetônicos constantes dos desenhos desenvolvidos nas etapas anteriores, com tratamento de acabamento e *rendering* (aplicação de materiais e iluminação);

Obtenção de imagens renderizadas do Modelo Geométrico Tridimensional. Ver APÊNDICE F.

Segundo Teles (2005), a análise do desempenho de um programa de computador objetiva [...] *estimar o tempo e os recursos consumidos, descobrir potenciais pontos de gargalo no processamento e prever o desempenho do sistema em um ambiente real.* (TELES, 2005, p. 29). Para isso é necessário testar a ferramenta em uso. Em seu estudo, ao desenvolver uma proposta para o plano de avaliação de desempenho de *software*, o autor utiliza uma abordagem baseada no modelo de qualidade em uso, definido na ABNT (2003a), contudo, restringiu-se à medição da eficiência do *software*, que é uma característica de qualidade externa, justificando que a mesma está relacionada com o desempenho do programa. Essa deliberação do autor corrobora com as prescrições da norma, uma vez que ela prevê que a qualidade do produto executável pode ser conferida pela combinação de atributos de qualidade externa, relacionados com características de qualidade em uso, conforme visto no capítulo 3. Partindo desse raciocínio construiu-se o modelo de qualidade apresentado no

quadro 16, especificando-se os atributos que constituem as características requisitadas das ferramentas selecionadas, consideradas nesta fase da avaliação proposta.

**Quadro 16** – Modelo de qualidade para avaliação dos programas livres gráficos

CARACTERÍSTICA DE QUALIDADE EM USO	CARACTERÍSTICA DE QUALIDADE EXTERNA	SUBCARACTERÍSTICA	ATRIBUTO
Eficácia	Funcionalidade	Acurácia	Capacidade de implementar as dimensões especificadas dos elementos arquitetônicos
		Adequação	Capacidade de implementar as tarefas especificadas
Produtividade	Eficiência	Comportamento em relação ao tempo	Capacidade de executar as tarefas especificadas no tempo especificado. Frequência de erros
Satisfação			Capacidade de satisfazer aos usuários

Fonte: Elaborado pelo autor, baseado em (ABNT, 2003a, p. 7 a 11).

Através do quadro 16 é possível identificar a relação feita entre as características de qualidade externa e as de qualidade em uso, sendo que os atributos desenvolvidos, a partir das subcaracterísticas, é que influenciaram na adoção, ou na construção, das métricas que foram aplicadas na medição de cada um desses atributos. Sodré (2006) chama atenção para o fato de que o usuário é quem faz o desdobramento das subcaracterísticas em atributos, de acordo com o interesse que o mesmo tem na avaliação do produto. A compilação do resultado da aplicação das métricas, sobre os dados coletados durante o experimento é que permitiu, ao final, a formulação de conclusões acerca do grau de desempenho da ferramenta perante os requisitos especificados. Observe-se que satisfação é uma característica de qualidade em uso que não foi vinculada a uma característica externa específica, como as demais, porque traduz a resposta do usuário à interação com o produto e inclui atitudes relacionadas ao seu uso (ABNT, 2003a, p. 12), o que torna difícil fazer a referida vinculação, já que depende também do perfil do usuário.

## 4.2 Ambientes de Aplicação do Experimento

A aplicação do experimento de avaliação foi executada nos seguintes locais, por dois profissionais de cada um deles:

- Laboratório de Pesquisa: Nesse laboratório, desenvolvem-se pesquisas e projetos arquitetônicos para edifícios e ambientes hospitalares e médicos, utilizando-se ferramentas computacionais, para gestão e produção, com base no emprego de *softwares* proprietários e, de forma mais incipiente, *softwares* livres. Para fins de identificação, os participantes do experimento no Laboratório foram denominados de P1 e P2.
- Instituição Pública Municipal: A função da Instituição é elaborar projetos de arquitetura e urbanismo para edifícios e espaços públicos da cidade. Conta com duas Gerências de Projetos: Especiais e de Arquitetura que, juntas, reúnem um “time” de 18 profissionais e quatro estagiários da área de AEC (Arquitetura, Engenharia e Construções), Artes Plásticas e Programação Visual. A rotina de trabalho desses profissionais envolve a concepção, desenvolvimento e detalhamento de projetos, com diversos graus de complexidade, utilizando sistemas gráficos computadorizados como principal ferramenta para representação, especificação e quantificação do projeto. Para fins de identificação, os participantes do experimento na Instituição foram denominados de P3 e P4.
- Escritório de Arquitetura: fundado em 1994, com ênfase na atuação em arquitetura de edifícios de saúde, área na qual os arquitetos titulares têm especialização, contudo, atuando também na elaboração de projetos residenciais, comerciais, corporativos e de arquitetura de interiores. A base de produção desse escritório é automatizada com auxílio de tecnologias computacionais. Para fins de identificação, os participantes do experimento no Escritório foram denominados de P5 e P6.

## 4.3 Tempo de Duração do Experimento

O experimento iniciou com o treinamento dos participantes na operação de recursos específicos das ferramentas selecionadas, suficiente para realização das tarefas e obtenção dos produtos requeridos na especificação dos requisitos de avaliação. Considerando que esse público possui conhecimentos suficientes, tanto na área de elaboração de projetos de baixa



complexidade, similar ao que foi aplicado na avaliação, quanto na operação de sistemas gráficos, evita-se que o conteúdo do treinamento aborde questões metodológicas referentes à elaboração de projetos, bem como conceitos básicos da tecnologia empregada nas ferramentas, que poderiam desviar o processo do objetivo deste trabalho. Ao final da etapa de treinamento, os usuários aplicaram os conhecimentos adquiridos em um exercício prático, que consistiu na representação do projeto do Módulo Policial citado anteriormente. Para dar apoio ao desenvolvimento das atividades de treinamento e execução do exercício prático, foi utilizado o seguinte material didático:

- Apresentação produzida no padrão .odp (Open Office) com ilustração dos principais elementos da interface dos programas;
- Lista de tarefas (Roteiro para Elaboração do Experimento) a serem cumpridas, passo a passo, para realização do exercício prático. Essa lista encontra-se no APÊNDICE C;
- Tabelas com valores das variáveis e dos parâmetros, requisitados pelos programas, para padronização da execução do exercício prático;
- Conjunto de peças gráficas, impressas, previamente elaboradas pelo avaliador, idênticas às que foram produzidas durante a avaliação. Uma cópia dessas peças consta dos APÊNDICES D, E e F.

Nessa etapa é que foi efetivada a execução da avaliação, propriamente dita, visando quantificar o desempenho das ferramentas ao mesmo tempo em que observou-se o processo de interação do usuário com o programa, suas reações, dificuldades, comentários e impressões, seguindo o viés de investigação da pesquisa qualitativa em um contexto específico de uso (NUNES, 1996). Dessa forma, o processo de avaliação, para cada uma das ferramentas, contemplou dois momentos

- Parte 1 – Treinamento dos usuários na utilização dos recursos do programa, utilizados para o cumprimento das tarefas especificadas no roteiro de elaboração do experimento. Essa etapa foi planejada com a seguinte carga horária: Caduntun (4h), Inkscape (4h) e Blender (12h).
- Parte 2 – Exercício prático de aplicação dos conhecimentos obtidos na etapa anterior, objetivando cumprir a ordem de tarefas listadas no “Roteiro para Elaboração do Experimento”, constante do APÊNDICE C, para fins de obter os produtos

especificados, sob supervisão do avaliador. Estimou-se que essa etapa teria a seguinte carga horária: Caduntun (4h), Inkscape (4h) e Blender (6h).

Dessa forma, somando-se a carga horária das partes 1 e 2, estimou-se que o tempo total de elaboração do processo de avaliação, em cada um dos ambientes de aplicação foi o seguinte: Caduntun (8h), Inkscape (8h) e Blender (18h). Essa estimativa foi formulada com base na experiência do avaliador no manuseio dos programas, e no tempo gasto pelo mesmo para elaboração prévia do experimento.

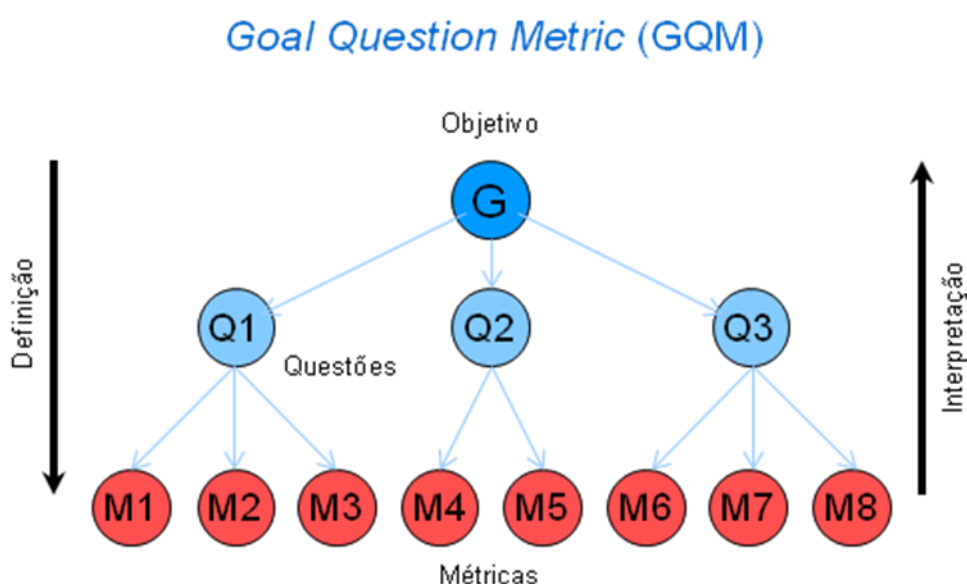
#### 4.4 Definição das Métricas

De acordo com o que foi visto no capítulo 3, as partes 2, 3 e 4 da ISO/IEC 9126, ainda não traduzidas para o idioma português, tratam respectivamente da correlação entre atributos e métricas, apresentando alguns exemplos aplicáveis em diversos casos, contudo, como observado anteriormente, apenas a parte 1 dessa norma foi traduzida para o idioma português pela ABNT. Sodré (2006) afirma que *Uma métrica pode ser definida como sendo toda particularidade quantificável do software que esteja relacionada a uma característica. Dependendo do produto analisado, diferentes métricas podem ser aplicadas.* (SODRÉ, 2006, p. 34). Da mesma forma que o usuário tem a liberdade para definir os atributos, de acordo com o interesse que tem no uso do programa, ele pode construir as métricas necessárias para medição desses atributos, desde que sejam pertinentes ao paradigma e à tecnologia adotada no procedimento de avaliação (SODRÉ, 2006). Em seu estudo, essa autora desenvolve uma abordagem sobre qualidade do produto de *software*, focada na descrição do modelo de qualidade da ISO/IEC 9126. No âmbito do seu trabalho, ela traduz algumas das tabelas de métricas externas, internas e de qualidade em uso, constantes das partes 2, 3 e 4 da norma, onde se pode verificar a existência de exemplos aplicáveis à avaliação do trabalho em curso.

Em seu estudo sobre elaboração de métricas no processo de desenvolvimento de *software*, Maria (2010) apresenta o método GQM (*Goal Question Metric*). De acordo com a autora, esse é um método que [...] *alinha as medições necessárias aos projetos de softwares com objetivos e metas da organização* [...] (MARIA, 2010, p.24). É um método simples, o qual, em resumo, consiste na definição de objetivos de qualidade que, posteriormente, são refinados em questões, das quais são extraídas métricas, que irão prover informações durante o processo de medição, para responder às questões levantadas e, então, verificar a situação da

organização em relação aos objetivos especificados, (MARIA, 2010). A figura 15 mostra uma representação esquemática da estrutura hierárquica GQM.

Esse método destaca-se pela facilidade com que pode ser aplicado e pela semelhança com a abordagem de qualidade da ISO/IEC 9126, uma vez que, nessa última, a definição das métricas segue a mesma lógica do GQM, ou seja, o propósito da métrica é definido sob forma de uma pergunta relacionada com o atributo de qualidade. Observa-se, também, que alguns autores utilizam métodos combinados das duas abordagens aqui referenciadas para avaliação de *software*, a exemplo de Kasburg (2001).



**Figura 15** – Estrutura do GQM

Fonte: (MARIA, 2010, p. 25).

Nesse contexto, a definição das métricas, para fins da avaliação das ferramentas selecionadas, baseou-se nos exemplos da ISO/IEC 9126, extraídos do trabalho de Sodr  (2006), e no m todo GQM, demonstrado por Maria (2010). Seguindo esse referencial, para cada um dos objetivos ou atributos de qualidade definidos, estabeleceu-se uma pergunta, ou conjunto de perguntas, que verifique o seu cumprimento. Com base nessas perguntas   que foram constru das as m tricas capazes de atribuir valor quantitativo  s respostas. O quadro 17 sintetiza esse processo e identifica as m tricas utilizadas para avalia o de cada um dos *softwares* selecionados. A defini o dessas m tricas considerou os atributos especificados no

quadro 16, elaborado anteriormente, sendo que a característica de satisfação deu origem a duas métricas: Taxa de adesão e Escala de satisfação; que foram avaliadas através do mesmo instrumento de medição: um questionário de satisfação respondido pelos participantes do experimento.

**Quadro 17 – Métricas propostas**

<b>OBJETIVO/ATRIBUTO</b>	<b>PERGUNTA</b>
Capacidade de implementar as dimensões especificadas dos elementos arquitetônicos	As dimensões especificadas estão implementadas?
Capacidade de implementar as tarefas especificadas	As tarefas especificadas foram implementadas com sucesso?
Capacidade de executar as tarefas especificadas no tempo especificado.	Quanto tempo se demora para executar uma tarefa?
Capacidade de evitar a ocorrência de erros	Qual a frequência de erros?
Capacidade de angariar usuários	Qual a proporção de usuários adotaria a ferramenta em suas atividades projetuais?
Capacidade de satisfazer os usuários	Qual o nível de satisfação do usuário?

Fonte: Elaborado pelo autor, baseado em Maria (2010, p. 39); Sodré (2006, p. 41 - 42).

Dando sequência à construção das métricas, criou-se o quadro 18, com base na metodologia adotada por Maria (2010) e nos exemplos de métricas constantes do trabalho de Sodré (2006). Recapitulando, a metodologia utilizada para esta fase da avaliação obedeceu à seguinte sequência de ações:

1. Definiu-se o objetivo/atributo com base nas prescrições da ABNT (2003a);
2. Questionou-se como atingir os objetivos através das perguntas;
3. Pesquisou-se no referencial teórico; (MARIA, 2010, p. 39) e (Sodré, 2006, p.41- 42) as métricas que poderiam responder às perguntas;

Quadro 18 – Métricas selecionadas

MÉTRICA	NOME	PROPÓSITO	MÉTODO DE APLICAÇÃO	MEDIDA E FÓRMULA	INTERPRETAÇÃO
1	Cobertura de implementação das dimensões	As dimensões especificadas estão implementadas?	Contar o número de dimensões implementadas com precisão e comparar com o número de dimensões especificadas	$X=A/B$ A = número de dimensões implementadas com precisão B= número de dimensões especificadas	$0 \leq X \leq 1$ Quanto mais próximo de 1, mais preciso
2	Adequação de implementação funcional	As tarefas especificadas foram implementadas com sucesso?	Contar o número de tarefas implementadas sem erros e comparar com o número de tarefas especificadas	$X = A/B$ A = número de tarefas implementadas sem erros B = número de tarefas especificadas	$0 \leq X \leq 1$ Quanto mais próximo de 1, mais adequado
3	Tempo de tarefa	Quanto tempo se demora para executar uma tarefa?	Contar o tempo gasto para realização das tarefas e comparar com o tempo estimado para essa atividade	$X = T$ T = tempo especificado para realização das tarefas	$X \geq T$ Quanto mais próximo de T, maior a eficiência
4	Frequência de erros	Qual é a frequência de erros?	Contar o número de falhas tomadas pelo usuário e comparar com o tempo total para efetivação das tarefas	$X = A/T$ A = número de falhas tomadas pelo usuário T = tempo ou número de tarefas	$0 \leq X$ Quanto mais próximo de 0 melhor
5	Taxa de adesão	Qual a proporção de usuários adotaria a ferramenta em suas atividades projetuais?	Contar o número de usuários que participaram do experimento e comparar com o número de usuários que manifesta intenção de adotar a ferramenta	$X= A/B$ A = número de usuários que manifestam intenção de adotar a ferramenta B = número de usuários que participam da avaliação	$0 \leq X \leq 1$ Quanto mais próximo de 1 melhor
6	Escala de satisfação	Qual o nível de satisfação do usuário?	Aplicar questionário de satisfação	$X = A$ A = resultado da pesquisa de satisfação	Comparação com a média da população entrevistada

Fonte: Elaborado pelo autor, baseado em Maria (2010, p. 42 - 44); Sodré (2006, p. 41– 44).

Identificadas as métricas, partiu-se para elaboração dos procedimentos de apuração (ABNT, 2001a). No caso do experimento realizado no âmbito deste estudo, o julgamento da avaliação das medições considera de forma prioritária o conceito a elas atribuídos, uma vez que, com

relação ao número de participantes do experimento, não se busca uma amostragem que atenda a questões estatísticas, mas sim ao viés de investigação da pesquisa qualitativa que, segundo Roratto (2010), baseia-se no processo dialógico interativo e relacional entre o pesquisador e os sujeitos da pesquisa, no caso os usuários dos programas, requerendo, portanto, uma amostragem pequena de participantes do experimento. O quadro 19 apresenta os níveis de pontuação e os respectivos conceitos a eles atribuídos.

**Quadro 19** – Níveis de Pontuação para fins de julgamento da avaliação

MÉTRICA	INTERPRETAÇÃO	NÍVEL DE PONTUAÇÃO / CONCEITOS			
		FRACO	MÉDIO	BOM	EXCELENTE
1 - Cobertura de implementação das dimensões	$0 \leq X \leq 1$ Quanto mais próximo de 1, mais preciso	$0 \geq 0,69$	$0,7 \geq 0,79$	$0,8 \geq 0,89$	$0,9 \geq 1,0$
2 - Adequação de implementação funcional	$0 \leq X \leq 1$ Quanto mais próximo de 1, mais adequado	$0 \geq 0,69$	$0,7 \geq 0,79$	$0,8 \geq 0,89$	$0,9 \geq 1,0$
3 - Tempo de tarefa	$X \geq 4$ $X \geq 6$ Quanto mais próximo do tempo especificado, melhor	$\geq 4h\ 31min$	$4h16min \geq 4h30min$	$4h1min \geq 4h15min$	$\geq 4h$
		$\geq 6h\ 31min$	$6h16min \geq 6h30min$	$6h1min \geq 6h15min$	$\geq 6h$
4 - Frequência de erros	$0 \leq X$ Quanto mais próximo de 0 melhor	$\leq 0,31$	$0,3 \leq 0,21$	$0,2 \leq 0,11$	$0,1 \leq 0$
5 - Taxa de adesão	$0 \leq X \leq 1$ Quanto mais próximo de 1 melhor	$0 \geq 0,69$	$0,7 \geq 0,79$	$0,8 \geq 0,89$	$0,9 \geq 1,0$
6 - Escala de satisfação	Comparação com a média da população entrevistada	Menos de 70 % de respostas afirmativas	Entre 70 e 79% de respostas afirmativas	Entre 80 e 89% de respostas afirmativas	Acima de 90% de respostas afirmativas

Fonte: Elaborado pelo autor com base em Sodré (2006, p. 42 - 44); ABNT (2001a, p.12).

Os níveis de pontuação estabelecidos nesse quadro, foram arbitrados pelo avaliador, levando-se em consideração os seguintes critérios:

- Para as métricas cuja interpretação admitiu-se que, quanto mais próximo de 1 for o

resultado da medição, maior é o nível de qualidade do *software*, considerou-se satisfatórios os resultados que alcançaram no mínimo 70% de pontuação, dentro da escala de medição;

- Para a métrica cuja interpretação admitiu-se que, quanto mais próximo de zero for o resultado da medição, maior é o nível de qualidade do *software*, considerou-se satisfatórios os resultados que alcançaram no máximo 30% de pontuação, dentro da escala de medição;
- No que se refere à métrica de nº 3, a especificação do nível de pontuação foi atrelada ao tempo estabelecido para conclusão da Etapa 2 do experimento - similares para o Caduntun e Inkscape (4h) e diferente para o Blender (6h) - durante a qual os participantes aplicaram os conhecimentos obtidos na etapa de treinamento, com vistas ao cumprimento da ordem de tarefas listadas no “Roteiro para Elaboração do Experimento”, objetivando obter os produtos nele especificados, sob supervisão do avaliador. Levou-se em conta que o tempo de tarefa não depende só do desempenho do *software*, mas também da habilidade do usuário, e dentro desta perspectiva considerou-se que os participantes da avaliação possuíam conhecimentos suficientes, tanto na área de elaboração de projetos de baixa complexidade, similar ao que foi desenvolvido, quanto na operação de sistemas gráficos, assim, admitiu-se como satisfatório o tempo de execução especificado para conclusão da Etapa 2 do experimento, com tolerância máxima de 30 minutos de acréscimo, dentro do nível de pontuação conceituado como médio.
- Com relação à métrica de nº 6, Escala de Satisfação, admitiu-se como satisfatórios os resultados que apresentaram um nível de repostas afirmativas, ao questionário aplicado, igual ou superior a 70% das perguntas elaboradas.

#### **4.5 Plano de Avaliação**

De acordo com a ABNT (2001a), o Plano de Avaliação, como o próprio nome diz, objetiva estabelecer o planejamento da avaliação prevendo os recursos necessários, métodos utilizados e o cronograma das ações do avaliador, auxiliando-o na condução da execução da avaliação. Trata-se também, especialmente no contexto desse trabalho, de um instrumento que irá

assegurar a repetibilidade do experimento, na medida em que especifica os procedimentos a serem adotados para execução da avaliação, apoiado pelas correspondentes características de qualidade em uso, definidas no quadro 16, elaborado anteriormente, e também pelos métodos e técnicas prescritos na etapa anterior de especificação. Para construção desse Plano, tomou-se como base o Modelo de Avaliação Quantitativa, constante do Anexo A (informativo) disponível na ABNT (2003b, p. 9). Observa-se que as tarefas-chaves que envolvem a projeção da avaliação e que resultará no Plano a que se pretende chegar, foram sintetizadas no quadro 6, constante do capítulo 3, e serão desenvolvidas a seguir.

#### **4.6 Seleção dos Participantes**

Para integrar a equipe do processo de avaliação foram selecionados dois participantes, em cada um dos ambientes escolhidos para aplicação do experimento. O recrutamento desses indivíduos foi feito com base no preenchimento das características de formação e atuação profissional constantes da especificação dos requisitos da avaliação, definida em 4.1, e na voluntariedade dos mesmos, observando-se a disponibilidade para cumprimento da carga horária estimada para as etapas de treinamento e execução do exercício prático, que resultou na elaboração dos produtos (plantas e peças gráficas) aos quais foram aplicadas as métricas para aferição dos resultados.

#### **4.7 Equipamentos Necessários à Avaliação e Fonte de Acesso às Ferramentas**

Para execução da Avaliação, os ambientes de execução do experimento dispunham de três computadores com, ao menos, a configuração mínima de *hardware* requerida pelos programas Caduntun, Inkscape e Blender, de forma que, cada uma dessas ferramentas podiam desempenhar todas as suas funcionalidades. Essas configurações dos equipamentos utilizados, inclusive a do ambiente de trabalho do avaliador, constam do APÊNDICE G. Dois desses equipamentos, entre os disponíveis em cada um dos ambientes de execução do experimento, destinaram-se ao uso dos participantes do treinamento/avaliação, e o terceiro ao uso do Avaliador, que foi o responsável pela capacitação dos usuários na operação dos programas e, também, pelo apoio e acompanhamento da execução do exercício prático que deu origem à avaliação. Para aquisição do arquivo de instalação dos programas fez-se o *download* dos mesmos através dos seguintes sítios da web:



- 1 – Caduntun: <<http://sourceforge.net/projects/librecad/files/>>
- 2 – Inkscape: <<http://wiki.softwarelivre.org/InkscapeBrasil/Download>>
- 3 – Blender: <<http://download.blender.org/release/Blender2.49b/>>

O primeiro reporta-se ao repositório do projeto do *software* Caduntun, uma vez que a página *web* de divulgação do programa possui *hiperlink* que direciona os visitantes interessados em fazer o *download* da ferramenta a esse sítio da internet (LIBRECAD, 2011). O segundo trata-se da sessão de *download* da página oficial da comunidade de desenvolvedores brasileiros do projeto [inkscape.org](http://inkscape.org); e o terceiro compõe a seção de *download* da página oficial de divulgação do projeto Blender, mantida pela Blender Foundation. A instalação dos *softwares*, em todos os computadores utilizados na realização do experimento, foi conduzida pelo Avaliador.

As versões dos programas avaliados foram as seguintes:

- Caduntun: Versão 1.0.0 beta 4, liberada em 21 de novembro de 2010;
- Inkscape: Versão 0.48.1, liberada em março de 2011,
- Blender: Versão 2.49b, liberada em 16 de setembro de 2009.

Com relação às duas primeiras, tratam-se das versões mais recentes dos programas, encontradas durante o tempo de elaboração desse estudo, entretanto, com relação ao Blender, optou-se por uma versão não atualizada, pelo fato de que, essa acima especificada, perdurou como última versão estável do *software* até o mês de abril de 2011, quando foi liberada sua sucessora, igualmente estável, a versão 2.57 (BLENDER FOUNDATION, 2011d). Contudo, observou-se, durante o período de investigação das ferramentas, que a principal mudança ocorrida, nessa nova versão, está na interface gráfica do programa, que busca ser mais “amigável” e atraente para o usuário.

#### **4.8 Metodologia de Condução da Avaliação**

Uma vez disponibilizados os locais de aplicação do experimento, equipados com os recursos materiais, *hardware* e *softwares*, conforme especificado, e estando disponíveis os recursos humanos— profissionais e estudantes voluntários selecionados entre o público alvo— estabeleceu-se em comum acordo com estes o cronograma para realização da etapa de

treinamento, cuja carga horária mínima foi a indicada anteriormente na especificação dos requisitos de avaliação e constam do APÊNDICE C. Os recursos necessários para as aulas foram os equipamentos relacionados no APÊNDICE G, programas e material didático já referenciados. As aulas, de caráter teórico e prático, seguiram uma metodologia similar para todos os programas em avaliação, consistindo na apresentação dos elementos da interface gráfica dos programas e treinamento no uso de suas funcionalidades, focando exclusivamente nas ferramentas mínimas necessárias para o cumprimento da ordem de tarefas explicitadas no "Roteiro para Elaboração do Experimento", constante do APÊNDICE C. Para isso, o Avaliador tirou partido de exemplos da aplicação dos recursos das ferramentas, em procedimentos similares aos que eram necessários para realização do exercício prático.

Foi levada em consideração a baixa complexidade do modelo proposto para execução da avaliação, e considerada também a fundamentação teórica das disciplinas de Projeto de Arquitetura, Desenho Arquitetônico e Desenho Técnico, para aplicação da didática necessária ao ensino da operação das ferramentas durante a fase de treinamento. Uma vez terminada essa fase, os participantes fizeram o exercício prático, buscando atender à carga horária estimada; 4 horas para o Caduntun, 4 horas para o Inkscape e 6 horas no caso do Blender. Quando necessário, entretanto, esse tempo foi extrapolado até a conclusão de todas as tarefas constantes do roteiro para elaboração do experimento. Durante a execução do exercício, coube ao Avaliador dar o devido acompanhamento aos trabalhos, sendo que a sua intervenção, nessa fase, restringiu-se às seguintes atividades:

- Anotar comentários, reações, eventuais dificuldades e problemas levantados pelos participantes durante a execução das tarefas;
- Registrar erros ou falhas no sistema;
- Cronometrar e controlar o tempo de execução das tarefas;
- Esclarecer dúvidas dos participantes com relação à operação das ferramentas, caso essas dúvidas representassem impedimento de continuidade ao desenvolvimento das tarefas.

O treinamento e a execução da avaliação foram feitos de forma sequencial e interdependente. Como o objetivo deste trabalho gira em torno da aplicação de um “pacote” composto de três *softwares* livres gráficos a um único modelo de projeto de arquitetura, os produtos extraídos

da experiência com o Caduntun, primeiro programa avaliado, foram utilizados como elementos de entrada para aplicação do experimento aos demais programas. Os arquivos padrão ".dxf"<sup>69</sup> gerados pelo Caduntun, foram convertidos ao padrão ".svg" antes de darem entrada no Inkscape, segundo *software* a ser avaliado. Para utilização no Blender, terceiro e último programa submetido ao experimento, os arquivos gerados no Caduntun não precisaram passar por processo de conversão, uma vez que, o Modelador importa arquivos gerados no padrão ".dxf".

Ao final de cada sessão, durante a execução do exercício prático, o avaliador gravou uma cópia dos arquivos resultantes, em fonte externa às máquinas utilizadas no experimento, visando preservar a inviolabilidade dos mesmos e certificar-se de que eles não seriam utilizados e (ou) alterados até a retomada dos trabalhos. Uma vez finalizados, os produtos gerados nessa segunda etapa do processo de avaliação, foi aplicado um Questionário de Satisfação aos participantes do processo. Os produtos resultantes do cumprimento das tarefas especificadas no "Roteiro para Elaboração do Experimento" foram recolhidos, em sua forma digital, para serem analisados pelo Avaliador em seu ambiente de trabalho. Esse material foi examinado utilizando-se as mesmas ferramentas empregadas no experimento, objetivando fazer as medições e tabular os dados que indicaram o nível de pontuação alcançado por cada um dos *softwares* livres gráficos, comparando-se os resultados encontrados à escala de pontuação definida no quadro 19. A forma de registro dos dados coletados foi efetivada através do preenchimento das "Fichas de Avaliação", criadas a partir da necessidade de sistematização dos dados requeridos para atender ao método de aplicação das métricas. Os modelos de "Ficha de Avaliação" utilizados encontram-se no APENDICE H.

#### **4.9 Execução da Avaliação**

Os procedimentos aplicados no experimento, foram feitos em conformidade com o Plano de Avaliação, fornecendo estimativas acuradas sobre o nível de atendimento às características de qualidade especificadas para cada uma das ferramentas avaliadas. Conforme colocado no início desse capítulo, uma vez concluída essa etapa dos trabalhos, foi feito o registro das informações acerca da execução da avaliação. Esse registro foi oriundo das anotações feitas pelo avaliador durante a execução do exercício prático e do preenchimento das "Fichas de

---

<sup>69</sup> O DXF – *Drawing ExchangeFormat*, é um arquivo de intercâmbio para modelos em CAD.

Avaliação”, reunindo os elementos necessários para, dessa forma, documentar os procedimentos realizados e os resultados obtidos em conformidade com a função do “Módulo de Avaliação de Qualidade”, prescrito pela ABNT (2004). Considerando que o escopo do experimento contemplou a aplicação do processo de avaliação, em separado, para cada um dos *softwares* livres gráficos em estudo (Caduntun, Inkscape e Blender), observa-se que os grupos de usuários selecionados, e os ambientes de execução, foram comuns a todos os contextos de avaliação. Levando-se em conta o preenchimento dos requisitos de avaliação definidos no item 4.1, foram selecionados dois participantes em cada uma das instituições escolhidas, cujo perfil pode ser conferido no questionário constante do APÊNDICE I. Esses participantes desenvolveram as atividades, utilizando equipamentos cujas configurações atendem ao especificado no Plano de Avaliação, tanto na etapa de treinamento, quanto na etapa de execução do exercício prático. A configuração desses equipamentos encontram-se descritas no APÊNDICE G

#### 4.10 Tempo da Avaliação

O tempo gasto por cada participante na execução das tarefas especificadas no “Roteiro para Elaboração do Experimento”, foi cronometrado pelo avaliador para fins de controle do cumprimento da carga horária especificada e registro dos casos em que foi necessário prolongar o prazo para execução do exercício prático. O quadro 14 apresenta o o registro do tempo despendido por cada participante para a elaboração e conclusão das tarefas envolvidas na elaboração do exercício prático.

**Quadro 20** – Tempo gasto por participante para execução do exercício prático

SOFTWARE	TEMPO ESPECIFICADO (h)	TEMPO GASTO (h:min)						MÉDIA (h:min)
		P1	P2	P3	P4	P5	P6	
CADUNTUN	4	02:37	03:32	03:22	03:45	03:51	05:37	03:47
INKSCAPE	4	02:10	02:10	01:47	02:05	02:52	04:42	02:37
BLENDER	6	04:50	05:40	05:27	05:59	06:05	08:48	06:08

Fonte: Elaborado pelo autor.

Observa-se nesse quadro que, das dezoito ocorrências registradas, apenas quatro extrapolaram o tempo especificado para conclusão da execução do exercício prático, sendo três dessas

ocorrências atribuídas a um único participante (P6), e duas ocorrências atribuídas a um único *software* (Blender). No que se refere ao tempo médio gasto com as tarefas, pode-se concluir que:

- o tempo médio gasto com a execução das tarefas realizadas com o Caduntun e o Inkscape foi abaixo do especificado;
- o tempo médio gasto com a execução das tarefas realizadas com o Blender extrapolou em oito minutos o tempo especificado,
- a extrapolação da média do tempo gasto para execução das tarefas realizadas com o Blender, foi decisivamente influenciada pelo desempenho de um único participante.

#### **4.11 Resultado das Tarefas**

Os produtos obtidos como resultado do procedimento de avaliação foram as peças gráficas especificadas no item 4.1, elaboradas pelos participantes durante a execução do experimento e entregues ao avaliador em meio digital, conforme requerido no Plano de Avaliação. A análise desse conteúdo consistiu no exame das informações gráficas e coleta dos dados necessários para aplicação das métricas especificadas no quadro 18, elaborado anteriormente. Os dados coletados foram sistematizados nas “Fichas de Avaliação” utilizadas para aferir os níveis de pontuação alcançados e apurar o quanto as ferramentas responderam aos requisitos da avaliação. Esse processo resultou no julgamento do grau de atendimento dos atributos de qualidade em uso especificados das ferramentas avaliadas, que será objeto de análise detalhada no item 4.14, que apresentará o resultado da aplicação das métricas e a respectiva interpretação dos mesmos através da “triangulação” dos dados quantitativos e qualitativos, conforme anunciado na introdução desse capítulo. A tabulação dos resultados referentes aos níveis de pontuação alcançados pelas ferramentas encontram-se nos quadros 27, 28 e 29 constantes do item acima referenciado.

#### **4.12 Resultados de Satisfação**

De acordo com a norma, o registro do resultado de satisfação se dá por meio do preenchimento de um questionário (ABNT, 2006, p. 29). Para fins deste estudo foi desenvolvido um questionário padronizado, o qual foi respondido por cada um dos participantes do experimento no contexto de avaliação de cada uma das ferramentas

selecionadas. O resultado obtido com o preenchimento foi sistematizado nos quadros 21 a 26, apresentados a seguir que serão brevemente comentados, reservadas análises mais detalhadas para o item 4.14, o modelo padronizado do questionário encontra-se no APÊNDICE J.

**Quadro 21** - Impressões dos participantes sobre o programa Caduntun

INTERFACE					EFICIÊNCIA RECURSOS					RESULTADOS				
Particip.	Ruim	Reg.	Bom	Ótimo	Particip.	Ruim	Reg.	Bom	Ótimo	Particip.	Ruim	Reg.	Bom	Ótimo
P1			X		P1			X		P1				X
P2			X		P2			X		P2				X
P3			X		P3			X		P3				X
P4		X			P4		X			P4			X	
P5		X			P5			X		P5				X
P6			X		P6		X			P6			X	

Fonte: Elaborado pelo autor.

**Quadro 22** - Graus de dificuldade atribuídos pelos participantes ao programa Caduntun

GRAU DE DIFICULDADE								ADESÃO AO PROGRAMA			
APRENDIZADO COMANDOS				APLICAÇÃO COMANDOS							
Particip.	Muito	Pouco	Nenhum	Particip.	Muito	Pouco	Nenhum	Particip.	Sim	Talvez	Não
P1			X	P1			X	P1	X		
P2		X		P2		X		P2	X		
P3		X		P3		X		P3	X		
P4		X		P4		X		P4			X
P5		X		P5		X		P5			X
P6		X		P6		X		P6		X	

Fonte: Elaborado pelo autor.

Fazendo uma análise combinada dos quadros 21 e 22, observa-se que, em geral, os resultados apresentados denotam um grau de satisfação elevado com os resultados alcançados a partir da aplicação do programa. Os usuários julgam que o *software* é de fácil aprendizado, possui recursos eficientes e uma interface amigável, de forma que houve pouca ou nenhuma dificuldade para aplicação dos comandos, embora dois dos participantes tenham manifestado que não adotariam a ferramenta em trabalhos futuros.

A leitura dos quadros 23 e 24 resulta na constatação de que a utilização do programa Inkscape produz resultados satisfatórios, na opinião dos usuários. A interface do programa é considerada boa ou ótima, assim como a eficiência dos recursos que foram empregados na

execução do exercício prático. Em sua maioria, os usuários tiveram pouca dificuldade no aprendizado e, conseqüentemente, não acharam difícil aplicar os recursos utilizados. Em decorrência dessa experiência os participantes se mostram com tendência a aderirem o programa em trabalhos futuros.

**Quadro 23** - Impressões dos participantes sobre o programa Inkscape

INTERFACE					EFICIÊNCIA RECURSOS					RESULTADOS				
Particip.	Ruim	Reg.	Bom	Ótimo	Particip.	Ruim	Reg.	Bom	Ótimo	Particip.	Ruim	Reg.	Bom	Ótimo
P1			X		P1			X		P1			X	
P2			X		P2			X		P2			X	
P3				X	P3				X	P3				X
P4				X	P4				X	P4				X
P5				X	P5			X		P5			X	
P6			X		P6			X		P6			X	

Fonte: Elaborado pelo autor

**Quadro 24** - Graus de dificuldade atribuídos pelos participantes ao programa Inkscape

GRAU DE DIFICULDADE								ADESÃO AO PROGRAMA			
APRENDIZADO COMANDOS				APLICAÇÃO COMANDOS							
Particip.	Muito	Pouco	Nenhum	Particip.	Muito	Pouco	Nenhum	Particip.	Sim	Talvez	Não
P1		X		P1			X	P1		X	
P2		X		P2		X		P2	X		
P3			X	P3			X	P3	X		
P4			X	P4			X	P4	X		
P5		X		P5		X		P5		X	
P6		X		P6		X		P6	X		

Fonte: Elaborado pelo autor.

**Quadro 25** - Impressões dos participantes sobre o programa Blender

INTERFACE					EFICIÊNCIA RECURSOS					RESULTADOS				
Particip.	Ruim	Reg.	Bom	Ótimo	Particip.	Ruim	Reg.	Bom	Ótimo	Particip.	Ruim	Reg.	Bom	Ótimo
P1		X			P1				X	P1			X	
P2		X			P2	X				P2			X	
P3		X			P3				X	P3				X
P4		X			P4				X	P4				X
P5		X			P5				X	P5			X	
P6		X			P6		X			P6			X	

Fonte: Elaborado pelo autor.

**Quadro 26** - Graus de dificuldade atribuídos pelos participantes ao programa Blender

GRAU DE DIFICULDADE								ADESÃO AO PROGRAMA			
APRENDIZADO COMANDOS				APLICAÇÃO COMANDOS							
Particip.	Muito	Pouco	Nenhum	Particip.	Muito	Pouco	Nenhum	Particip.	Sim	Talvez	Não
P1	X			P1	X			P1		X	
P2	X			P2	X			P2		X	
P3	X			P3	X			P3	X		
P4		X		P4		X		P4	X		
P5	X			P5		X		P5	X		
P6		X		P6		X		P6		X	

Fonte: Elaborado pelo autor.

Os resultados apresentados nos quadros 25 e 26 demonstram que a experiência de utilização do Blender produziu, em geral, com apenas uma exceção, boas impressões aos participantes do experimento, no que se refere aos recursos do programa e aos resultados alcançados. Contudo, as impressões sobre a interface do programa não são boas e, conseqüentemente, se desdobram em um alto grau de dificuldade para o aprendizado e aplicação dos comandos. Não obstante, todos os participantes do experimento manifestam a tendência em adotar o programa para trabalhos futuros.

#### 4.13 Dificuldades encontradas

Durante a aplicação do experimento, envolvendo a fase de treinamento e de execução do exercício prático, algumas dificuldades surgiram decorrentes de hábitos adquiridos pelos usuários, como reflexo da consolidação do uso de um universo ainda restrito de *softwares* gráficos proprietárias, que se consolidaram como padrão de ferramenta computacional aplicada na automação do projeto, tanto no ensino, quanto na prática da arquitetura. Essa afirmação pode ser feita com base nas respostas dadas no questionário constante do APÊNDICE I, Através desse questionário é possível observar que os programas utilizados na rotina profissional e estudantil desses indivíduos restringem-se a um mesmo conjunto de ferramentas, largamente difundidas e utilizadas. Essa ordem de dificuldades se manifestaram de forma mais contundente relacionadas com a utilização do Caduntun e do Blender, devido aos seguintes aspectos:



## 1 – Caduntun:

- Apesar de possuir uma interface gráfica similar à de programas CAD utilizados pelos participantes do experimento, como o AutoCAD, a forma de interação do usuário com o programa difere na execução dos seguintes comandos testados:
  - Desfazer seleção de entidades gráficas
  - Operação do *Snap*
  - Forma de fazer a execução de comandos de edição de desenhos

A forma de interagir com essas funcionalidades do programa requer um número maior de "passos" do que os exigidos pelos programas utilizados na rotina dos participantes do curso, causando estranheza e erros ao se tentar interagir com esses comandos de forma idêntica aos de uso habitual.

## 2 – Blender:

- Interface gráfica muito diferenciada dos padrões adotados pelos programas de modelagem, *rendering* e animação utilizados pelos participantes do experimento;
- Quantidade elevada de comandos "ocultos", executados por teclas de atalho, em vez da utilização de ícones na interface gráfica e, muitas vezes, quando representados no ecrã estão sob a forma de botões com informações textuais, configurando painéis de comando que não sugerem uma relação visual imediata de suas funcionalidades, a exemplo dos ícones, frequentemente utilizados em programas difundidos na prática da arquitetura.

No que se refere à utilização do Inkscape, detectou-se outra ordem de dificuldade relacionada com problemas de impressão. Neste caso, para a impressão de desenhos com áreas preenchidas com padrões de texturas importadas para dentro do programa, foi necessário abrir o arquivo em outro programa livre (Open Office), converter o arquivo para o formato de imagem e proceder a impressão sem erros. Como a interface gráfica do Inkscape e a forma de interação com os comandos é bastante parecida com a do CorelDraw, programa utilizado por todos os participantes do experimento, a exceção de apenas um deles, mas que também é um programa bastante difundido nos meios acadêmicos e profissionais da arquitetura, não foram registrados problemas relacionados com a apreensibilidade do *software*.

#### 4.14 Métricas e Medidas

A metodologia adotada para seleção dos *softwares* livres gráficos, dentro do universo pesquisado, levou à escolha dos programas Caduntun, Inkscape e Blender como sendo aqueles que reúnem o maior potencial de atendimento às características de qualidade especificadas para realização da avaliação objetivada neste estudo. As métricas propostas e as medições realizadas, tendo como base o resultado das tarefas executadas pelos participantes durante o processo de avaliação, denotaram o nível de pontuação alcançado pelos *softwares* avaliados com relação ao atendimento das características de qualidade em uso. Não obstante, conforme visto no segundo capítulo, e aqui confirmado, o desempenho de um programa gráfico está atrelado às suas condições de uso, na medida em que faz parte de um sistema composto de recursos da informática e de recursos humanos que com ele interage, e como tal, a qualificação desses últimos interfere sobremaneira no desempenho dessas ferramentas (NUNES, 1997). O público alvo do experimento realizado foi formado por indivíduos que possuem o perfil especificado para a avaliação, passaram pelo mesmo processo de treinamento e executaram o exercício prático atendendo a parâmetros padronizados, contudo, a prerrogativa da composição do sistema computadorizado de auxílio ao projeto, levou à ponderação de aspectos relacionados com os diferentes níveis de capacitação e maturidade desses indivíduos. O tamanho reduzido da amostragem de participantes deveu-se ao fato de que, para os fins deste trabalho, a questão estatística não assume relevância por se tratar de uma pesquisa cuja avaliação é de caráter eminentemente qualitativo e, dentro desse viés de investigação, o emprego de métodos quantitativos teve como finalidade congregação e identificação de variáveis específicas à visão global qualitativa, possibilitando o enriquecimento das constatações obtidas sob condições controladas (NEVES, 1996). Buscou-se nos procedimentos normativos de avaliação de *software* prescritos pela ABNT o foco na qualidade em uso do produto, como reflexo de sua qualidade interna, ou seja, do seu ciclo de desenvolvimento, e da sua qualidade externa, como resposta da interação dos programas com os componentes de *hardware*, para a produção de dados quantitativos que, confrontados com aspectos da investigação focada no preparo, nas dificuldades, bem como nas reações individuais de cada um dos participantes do experimento, observados durante a interação deles com os programas avaliados, e manifestadas como reflexo do perfil de cada um, formou-se a base de subsídios necessários para proceder a "triangulação" dos dados quantitativos e qualitativos que apresenta-se como resultado da avaliação, desenvolvido a seguir.

**Quadro 27 –** Tabulação dos resultados da medição das características de qualidade do Caduntun

CADUNTUN													
PONTUAÇÃO ATRIBUÍDA A CADA UM DOS CRITÉRIOS DE AVALIAÇÃO DA QUALIDADE EM USO										NÍVEL DE PONTUAÇÃO			
NOME	MEDIDA E FÓRMULA	MEDIÇÃO	P1	P2	P3	P4	P5	P6	MÉDIA	FRACO	MÉDIO	BOM	EXCELENTE
Cobertura de implementação das dimensões	X=A/B A= número de dimensões implementadas com precisão B = número de dimensões especificadas	A	10	9	10	10	11	9	9,83	0,7>=0,69	0,7>=0,79	0,8>=0,89	0,9>=1,0
		B	12	12	12	12	12	12	12				
	MEDIDA	<b>0,83</b>	<b>0,75</b>	<b>0,83</b>	<b>0,83</b>	<b>0,92</b>	<b>0,75</b>	<b>0,82</b>					
Adequação de implementação funcional	X = A/B A = número de tarefas implementadas sem erros B = número de tarefas especificadas	A	61	58	61	58	59	47	57,33	0,7>=0,69	0,7>=0,79	0,8>=0,89	0,9>=1,0
		B	67	67	67	67	67	67	67				
	MEDIDA	<b>0,91</b>	<b>0,87</b>	<b>0,91</b>	<b>0,87</b>	<b>0,88</b>	<b>0,7</b>	<b>0,86</b>					
Tempo de tarefa	X >= T Quanto mais próximo de T maior a eficiência	T	02:37*	03:32*	03:22*	03:45*	03:51*	05:37	03:47*	>= 4h 31min	4h 16min>=4h 30min	4h 1min>=4h 15min	>= 4h
		MEDIDA	>= 4h	>= 4h	>= 4h	>= 4h	>= 4h	>= 4h 31min	>= 4h				
Frequência de erros	X = A/T A = número de falhas tomada pelo usuário T = número de tarefas	A	0	0	0	0	0	0	0	=< 0,31	0,3 =< 0,21	0,2 =< 0,11	0,1 =< 0
		T	61	58	61	58	59	47	57,33				
		MEDIDA	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>				
Taxa de adesão	X = A/B A = número de usuários que manifestaram intenção de adotar a ferramenta B = número de usuários que participaram da avaliação	A	4						0 <= X <= 1 Quanto mais próximo de 1 melhor	0,7>=0,69	0,7>=0,79	0,8>=0,89	0,9>=1,0
		B	6										
		MEDIDA	<b>0,67</b>										
Escala de satisfação	X = A A = resultado da pesquisa de satisfação	% Respostas Afirmativas	100	100	100	50	83	83	86	Menos de 70% de respostas afirmativas	Entre 70% e 79% de respostas afirmativas	Entre 80% e 89% de respostas afirmativas	Mais de 90% de respostas afirmativas

\* Tempo inferior ao mínimo estimado para o tempo de tarefa

Fonte: Elaborado pelo autor..

### a) Resultado do Caduntun

Observando-se o quadro 27 conclui-se que o Caduntun tem bom nível de implementação das dimensões, bem como de adequação funcional. As tarefas foram executadas, em média, abaixo do tempo estimado atingindo, neste quesito, grau de excelência acima do esperado, sem nenhuma ocorrência de falhas do programa, ou do sistema, gerando um bom nível de satisfação ao usuário, contudo com fraco desempenho no que se refere ao nível de adesão. Considerando a relevância que a implementação das dimensões tem para o projeto de arquitetura, cabe ressaltar que observou-se, durante a execução do exercício prático, que as dimensões implementadas com erro foram consequência da inserção de parâmetros incorretos por parte do usuário que, em geral, sentiu dificuldade para operar as funcionalidades de *Snap* do programa. Com base na análise do perfil dos participantes, nos resultados do questionário de satisfação e no acompanhamento dado durante a execução do experimento, foi possível constatar ainda que o baixo nível de dificuldade registrado e a boa impressão causada pela interface gráfica do programa se dão pela similaridade existente com a interface de programas CAD proprietários de uso genérico, que seguem o padrão difundido pelo AutoCAD, da *Autodesk*, adotado na prática profissional pelos participantes do experimento. Essa similaridade está presente na disposição dos comandos, no formato da área de desenho, na apresentação dos ícones e na nomenclatura dos elementos do menu. Contudo, os aspectos que conferiram dificuldade, ainda que pouca, na aprendizagem e na aplicação dos comandos do *software*, observados durante o treinamento e a execução do experimento, foram detectados principalmente nas seguintes operações:

- modo de seleção dos objetos e interação com comandos de edição de desenhos;
- manipulação dos comandos de precisão (*snap*);
- implementação do dimensionamento,
- inserção de linhas paralelas.

Constatou-se que, apesar do Caduntun ter atingido resultados satisfatórios em meio a uma interface amigável e bem difundida, os procedimentos necessários para realização dessas e outras operações, diferem do padrão usual. Esse aspecto também influenciou na baixa taxa de adesão, na medida em que ocorreram manifestações de comparação, e preferência, a programas produzidos em escala comercial, devido ao maior número de recursos neles existentes, que dão maior nível de automação à realização de operações como as supracitadas, sendo que, não se levantou relevância às implicações legais que o uso de cópias “piratas” desses programas traz para o usuário.

**Quadro 28** – Tabulação dos resultados da medição das características de qualidade do Inkscape

INKSCAPE													
PONTUAÇÃO ATRIBUÍDA A CADA UM DOS CRITÉRIOS DE AVALIAÇÃO DA QUALIDADE EM USO										NÍVEL DE PONTUAÇÃO			
NOME	MEDIDA E FÓRMULA	MEDIÇÃO	P1	P2	P3	P4	P5	P6	MÉDIA	FRACO	MÉDIO	BOM	EXCELENTE
Cobertura de implementação das dimensões	$X=A/B$ A = número de dimensões implementadas com precisão B = número de dimensões especificadas	A	9	10	10	10	11	9	9,83	0,7>=0,69	0,7>=0,79	0,8>=0,89	0,9>=1,0
		B	11	11	11	11	11	11	11				
		MEDIDA	0,82	0,91	0,91	0,91	1	0,82	0,89				
Adequação de implementação funcional	$X = A/B$ A = número de tarefas implementadas sem erros B = número de tarefas especificadas	A	50	49	51	54	54	48	51	0,7>=0,69	0,7>=0,79	0,8>=0,89	0,9>=1,0
		B	54	54	54	54	54	54	54				
		MEDIDA	0,93	0,91	0,94	1	1	0,89	0,94				
Tempo de tarefa	$X \geq T$ Quanto mais próximo de T maior a eficiência	T	02:10*	02:10*	01:47*	02:05*	02:52*	04:42	02:37*	>= 4h 31 min	4h 16min>=4h 30min	4h 1 min>=4h 15min	>= 4h
		MEDIDA	>= 4h	>= 4h	>= 4h	>= 4h	>= 4h	>= 4h 31 min	>= 4h				
Frequência de erros	$X = A/T$ A = número de falhas tomada pelo usuário T = número de tarefas	A	2	1	1	0	0	1	0,83	<= 0,31	0,3 <= 0,21	0,2 <= 0,11	0,1 <= 0
		T	50	49	51	54	54	48	51				
		MEDIDA	0,04	0,02	0,02	0	0	0,02	0,02				
Taxa de adesão	$X = A/B$ A = número de usuários que manifestaram intenção de adotar a ferramenta B = número de usuários que participaram da avaliação	A	6						0 <= X <= 1 Quanto mais próximo de 1 melhor	0,7>=0,69	0,7>=0,79	0,8>=0,89	0,9>=1,0
		B	6										
		MEDIDA	1										
Escala de satisfação	$X = A$ A = resultado da pesquisa de satisfação	% Respostas Afirmativas	100	100	100	100	100	100	100	Menos de 70% de respostas afirmativas	Entre 70% e 79% de respostas afirmativas	Entre 80% e 89% de respostas afirmativas	Mais de 90% de respostas afirmativas

\* Tempo inferior ao mínimo estimado para o tempo de tarefa.

Fonte: Elaborado pelo autor

## b) Resultado do Inkscape

De posse dos resultados da avaliação do Inkscape, apresentados no quadro 28, concluiu-se que esse programa reúne, em níveis muito elevados, os atributos requisitados no âmbito da especificação do experimento, de forma que, nesse contexto, atingiu um alto grau de satisfação das características de qualidade em uso dele requisitadas, destacando-se o tempo de tarefa, em média, inferior ao mínimo estipulado. O experimento de avaliação propôs a utilização de um "pacote" de *software* livre gráfico para o desenvolvimento de etapas de um mesmo projeto de arquitetura, assim, o exercício efetuado com o uso do Inkscape utilizou os arquivos gerados no Caduntun como dado de entrada e, dessa forma, os erros de implementação das dimensões foi herdado do outro *software*, mas, a operação de escalonamento do desenho nele inserido ocorreu com a precisão desejada. Relativo à adequação funcional, o Inkscape atingiu a excelência, entretanto, foi o único entre os programas avaliados que registrou ocorrência de erros tomados pelo usuário, não tendo sido possível detectar se foi erro do sistema ou do programa. Contudo, essas ocorrências não impactaram negativamente na taxa de adesão ao *software*. Constatou-se que a ampla similaridade, das funcionalidades, da aparência, e do modo de interação da interface gráfica do Inkscape com a do Corel Draw, programa líder de mercado no segmento de editores gráficos vetoriais, contribuiu sobremaneira para esse alto nível de aceitação da ferramenta.

Depois de uma análise do perfil dos usuários e do questionário de satisfação, concluiu-se que aqueles participantes que consideraram os resultados ótimos são os mais experientes na utilização de programas gráficos e possuem os níveis mais elevados de domínio do Corel Draw, bem como observou-se durante a execução do experimento, que esses indivíduos demonstravam maior autonomia, arriscando-se com sucesso na exploração de comandos que ainda não tinham sido explorados no treinamento. Mesmo aqueles participantes com conhecimento básico em programas desse tipo, não apresentaram dificuldade significativa no aprendizado dos comandos do Inkscape, opinando pelo bom resultado referente ao nível de eficiência dos recursos utilizados. Esses fatos levam a concluir que o Inkscape possui um grande potencial de aceitação por profissionais de arquitetura que fazem uso dessa categoria de programa. Não obstante, as dificuldades encontradas com a questão da impressão dos desenhos que traziam texturas importadas de fonte externa como padrão de preenchimento causou desânimo momentâneo, até que fosse encontrada uma solução para o problema, ficando a impressão de desvantagem do programa em relação ao seu equivalente proprietário.

**Quadro 29** – Tabulação dos resultados da medição das características de qualidade do Blender

BLENDER													
PONTUAÇÃO ATRIBUÍDA A CADA UM DOS CRITÉRIOS DE AVALIAÇÃO DA QUALIDADE EM USO										NÍVEL DE PONTUAÇÃO			
NOME	MEDIDA E FÓRMULA	MEDIÇÃO	P1	P2	P3	P4	P5	P6	MÉDIA	FRACO	MÉDIO	BOM	EXCELENTE
Cobertura de implementação das dimensões	$X=A/B$ A = número de dimensões implementadas com precisão B = número de dimensões especificadas	A	9	9	9	9	10	8	9	0,7 >= 0,69	0,7 >= 0,79	0,8 >= 0,89	0,9 >= 1,0
		B	10	10	10	10	10	10	10				
		MEDIDA	0,9	0,9	0,9	0,9	1	0,8	0,9				
Adequação de implementação funcional	$X = A/B$ A = número de tarefas implementadas sem erros B = número de tarefas especificadas	A	49	46	49	49	49	29	45,17	0,7 >= 0,69	0,7 >= 0,79	0,8 >= 0,89	0,9 >= 1,0
		B	53	53	53	53	53	53	53				
		MEDIDA	0,92	0,87	0,92	0,92	0,92	0,55	0,85				
Tempo de tarefa	$X \geq T$ Quanto mais próximo de T maior a eficiência	T	04:50*	05:40*	05:27*	05:59*	06:05	08:48	06:08	$\geq 6h 31min$ $6h 16min \geq 6h 30min$ $6h 1min \geq 6h 15min$	$\geq 6h 31min$ $6h 16min \geq 6h 30min$ $6h 1min \geq 6h 15min$	$\geq 6h 31min$ $6h 16min \geq 6h 30min$ $6h 1min \geq 6h 15min$	$\geq 6h$
		MEDIDA	$\geq 6h$	$\geq 6h$	$\geq 6h$	$\geq 6h$	$6h 1min \geq 6h 15min$	$\geq 6h 31min$	$6h 1min \geq 6h 15min$				
Frequência de erros	$X = A/T$ A = número de falhas tomada pelo usuário T = número de tarefas	A	1	2	2	0	1	3	1,5	$\leq 0,31$	$0,3 \leq 0,21$	$0,2 \leq 0,11$	$0,1 \leq 0$
		T	49	46	49	49	49	29	45,17				
		MEDIDA	0,02	0,04	0,04	0	0,02	0,1	0,04				
Taxa de adesão	$X = A/B$ A = número de usuários que manifestaram intenção de adotar a ferramenta B = número de usuários que participaram da avaliação	A	6						$0 \leq X \leq 1$ Quanto mais próximo de 1 melhor	0,7 >= 0,69	0,7 >= 0,79	0,8 >= 0,89	0,9 >= 1,0
		B	6										
		MEDIDA	1										
Escala de satisfação	$X = A$ A = resultado da pesquisa de satisfação	% Respostas Afirmativas	50	50	50	83	50	67	58,33	Menos de 70% de respostas afirmativas	Entre 70% e 79% de respostas afirmativas	Entre 80% e 89% de respostas afirmativas	Mais de 90% de respostas afirmativas

\* Tempo inferior ao mínimo estimado para o tempo de tarefa

Fonte: Elaborado pelo autor.

### c) Resultado do Blender

Diante dos resultados apresentados no quadro 29, foi possível concluir que, no contexto de uso especificado para realização do experimento, o Blender apresentou baixo nível de satisfação do usuário, embora as características de uso relacionadas com a funcionalidade e com a eficiência do programa tenham alcançado níveis satisfatórios, levando a um excelente nível de intenção de adesão ao programa.

Com base na análise dos questionários de satisfação, do perfil dos usuários e das observações feitas durante o processo de aplicação do experimento, foi possível concluir que a interface gráfica do Blender foi um dos principais fatores que contribuíram para esse resultado controverso, pois esteve diretamente relacionada com as dificuldades observadas durante o treinamento dos usuários para uso dos comandos do programa, devido, entre outros fatores, ao número elevado de funções acionadas por meio do teclado, e uma baixa taxa de elementos de interação baseado em ícones, muito comuns em programas dessa natureza, exigindo do usuário a memorização de sequências, às vezes complexas de procedimentos para efetivação de alguns comandos, a exemplo de operações booleanas de união, interseção e subtração de volumes, muito empregadas na construção de formas complexas, frequentes em projetos de arquitetura. Esses aspectos geraram dificuldades até mesmo naqueles usuários com maior capacitação no domínio de ferramentas de modelagem geométrica tridimensional. Observou-se que dada, também, a baixa complexidade do projeto utilizado como modelo para a avaliação, uma vez assimilada a forma de interação com o programa, a maioria dos participantes conseguiu executar o exercício dentro do tempo especificado, havendo apenas um caso em que a extrapolação desse tempo foi muito superior ao nível de tolerância também especificado, justamente por parte do usuário que não tinha nenhuma experiência com o uso de ferramentas CAD 3D e demonstrava alguma dificuldade com a própria lógica do raciocínio tridimensional, no que se refere ao senso de localização, translação e rotação de objetos no espaço. Observou-se que a interação com o programa ficava mais fácil na medida em que se dominava as operações relacionadas com a seleção dos objetos, movimentação, e edição, a partir daí o tempo gasto para a tomada de decisões acertadas variou de acordo com o nível de desenvolvimento da visão tridimensional do indivíduo. Com base nessas observações, constatou-se que o Blender é um programa que demanda muita prática para que se ganhe “intimidade” com ele e se memorize a forma relativamente complexa e peculiar de interagir com seus comandos. Os participantes do experimento mais experientes no uso de ferramentas



CAD 3D admitiram que uma vez superadas as dificuldades de aprendizado, o programa não parece tão complexo e oferece os atributos necessários para atendimento às características de qualidade em uso especificadas.

## **5. CONCLUSÃO**

Face aos objetivos traçados para este estudo, os procedimentos metodológicos empregados para alcançá-los abordaram aspectos conceituais e históricos do *software* livre, trazendo a tona aspectos filosóficos, motivações, forma de produção e acesso a essas ferramentas, ao tempo em que buscou o entendimento do processo de automação do projeto, as ferramentas e tecnologias empregadas e tendências desse processo, como forma de munir a pesquisa de uma base teórico-conceitual, que se valeu de técnicas e métodos normativos para selecionar, em um amplo universo de projetos de *software* livres gráficos, aquelas ferramentas com maior potencial de aplicação em projetos de arquitetura, sejam elas; o Caduntun, o Inkscape e o Blender, consideradas as características específicas do contexto de uso desses programas, na forma de um estudo de caso. A abordagem da pesquisa, de cunho prioritariamente qualitativo, agregou métodos quantitativos como forma de congregar controle e repetibilidade aos procedimentos adotados e, assim, reafirmar a validade e a confiabilidade dos resultados alcançados, apresentados anteriormente e, que, agora, subsidiam a formulação das conclusões que se seguem e de recomendações para o processo de desenvolvimento dessas ferramentas, com vistas ao atendimento das expectativas do processo de automação do projeto.

Com base nos resultados da avaliação, concluiu-se, de forma geral, que os programas avaliados podem ser empregados na prática de projetos de arquitetura, em etapas e graus de complexidade diferenciados, de acordo com a aplicabilidade de cada um deles. De forma mais específica, chegou-se às seguintes conclusões:

### **Caduntun**

Com a característica de ser um programa CAD 2D, o *software* respondeu satisfatoriamente ao contexto da avaliação, demonstrando que tem bom potencial para ser utilizado na produção de desenhos técnicos em projetos de arquitetura de baixa complexidade. Os resultados alcançados impactaram positivamente os participantes do experimento prático pelo nível de atendimento às tarefas especificadas, entretanto falta-lhe, na visão do usuário, agregar recursos que melhorem a usabilidade no que se refere aos comandos de seleção de objetos, manipulação do *snap*, implementação de cotas e interação com os comandos de edição, uma vez que, comparado ao AutoCAD, largamente utilizado pelos participantes do experimento, requer uma sequência maior de “passos” para a correta manipulação desses recursos. Confrontando-se o seu atual estágio de desenvolvimento com o levantamento feito no capítulo 2, acerca da trajetória evolutiva da tecnologia CAD, pode-se notar que a impressão é de que o

Caduntun apresenta um certo “atraso” com relação às possibilidades abertas no campo da tecnologia da informática aplicada ao projeto, considerando o desenvolvimento das linguagens de programação, o incremento de constante do poder de processamento dos computadores e a incorporação de outras tecnologias à tecnologia CAD. Não obstante esses aspectos, rapidamente absorvidos pela indústria do *software* proprietário, e que no processo de desenvolvimento do *software* livre depende da comunidade que se consiga agregar em torno do desenvolvimento da ferramenta, o Caduntun se apresenta como alternativa potencialmente aplicável na representação para trabalhos escolares, projetos arquitetônicos de baixa complexidade e como ferramenta de ensino dos princípios da informática aplicada ao projeto.

### **Inkscape**

O programa editor gráfico de vetores é, entre os *softwares* avaliados, aquele que mais se aproxima dos padrões estabelecidos no mercado, tanto no que se refere às suas funcionalidades, quanto pela interface gráfica, contudo, possui ainda problemas referentes à sua qualidade externa, ou seja, na interação do *software* com o sistema, manifestado pelos problemas apresentados com a impressão dos trabalhos, em todos os ambientes no qual foi submetido à avaliação, denotando, dessa forma, problemas com o seu desenvolvimento, contornados de forma considerada satisfatória, contudo, com solução que envolve a aplicação de outro *software* livre. A alta taxa de adesão ao programa se deu como reflexo dos resultados alcançados, precisos e totalmente dentro das especificações, sem lograr muito esforço do usuário pela apreensibilidade facilitada do programa, devido à sua similaridade com o Corel Draw, largamente utilizado para a produção de desenhos de apresentação no meio profissional de arquitetura. No tocante a esse aspecto, a similaridade do *software* gráfico livre com programas proprietários de grande difusão nos meios profissionais, assume importância fundamental na sua aceitação, no que se pôde observar, pelo fato de o usuário buscar sempre o “menor caminho”, tanto para o aprendizado quanto para a realização das tarefas envolvidas no trabalho e, se esse caminho pode ser intuitivo, aumenta a sua atratividade com relação ao programa. Nesse contexto, o Inkscape segue uma linha de desenvolvimento que o leva a ser uma promissora alternativa de acesso a programas gráficos aplicados ao projeto de arquitetura, uma vez que os problemas apresentados não influenciaram de forma significativa na sua aceitação pelo grupo de usuários envolvidos na avaliação, que passaram a impressão de satisfação em dispor de uma ferramenta capaz de substituir aquela a que não lhes parecia haver alternativa, confirmada essa impressão através dos dados quantitativos do experimento.

## Blender

No âmbito do contexto de uso em que o *software* foi selecionado e avaliado, ficou evidente que o Blender possui os principais atributos funcionais que caracterizam uma ferramenta de edição geométrica tridimensional para a construção de modelos digitais 3D em arquitetura, tendo sido o único programa livre gráfico, entre os avaliados, que possui histórico expressivo de aplicação nesta área. A credibilidade no *software* ganhou força devido à estrutura estável e bem organizada do seu projeto de desenvolvimento, suportado financeiramente por um modelo de negócio baseado no fornecimento de serviços e suporte técnico, que atingiu a dimensão necessária para retro-alimentar o investimento em tecnologia de programação, para que ele se mantenha no páreo com os concorrentes proprietários, no que se refere a seus recursos e potencial de aplicação. Contudo, no contexto da avaliação realizada, observou-se que o *software* impressiona, mas não conquista o usuário comum, acostumado com padrões de interface gráfica baseada em ícones. Quando estes se deparam com um programa estruturado para trabalhar preferencialmente através do uso combinado de teclas de atalho e painéis repletos de botões – que, em verdade, carregam muitas funcionalidades, mas requerem um esforço grande de memorização para apreensão da localização dos comandos e da sequência necessários à sua execução – desenvolvem um certo preconceito relativo ao programa, considerado de difícil apreensão. Alguns desistiriam sem nem mesmo ter tentado. Diante desse aspecto, o *software* requer treinamento assistido para aqueles que não dominam, ou não utilizam com frequência, ferramentas de modelagem tridimensional e, por parte daqueles que já possuem essa prática, se prevê um período prolongado de adaptação. Não obstante, esses últimos, depois de entenderem a lógica da interface do programa, passam a intuir certas funcionalidades e começam a desfazer as barreiras construídas nas primeiras impressões, valorizando a ferramenta e atribuindo-lhe o mérito de ser um programa com recursos avançados para modelagem, *rendering* e animação, equivalentes a de programas proprietários consolidados nos meios profissionais de arquitetura, como o 3Ds Max, da Autodesk, com grande portabilidade, fácil instalação e, o melhor, gratuito.

Ressalta-se, entretanto, que todos esses programas encontram barreiras à sua disseminação na medida em que são comparados a programas comerciais proprietários, que se consolidaram como ferramentas de automação do projeto pelo fato de que a tecnologia neles empregada evoluiu par e passo com a tecnologia computacional, influenciando a própria forma de se projetar em arquitetura.

Demonstrou-se, neste trabalho, o avanço alcançado pelos sistemas CAD proprietários, que passaram a ser utilizados maciçamente e a fazer parte do cotidiano profissional de empresas e particulares. Esse avanço foi alavancado à base de muito investimento em pesquisa, fomentado pela lógica do mercado. No caso dos *softwares* livres gráficos, conclui-se, com base no estudo realizado, que o avanço de seu desenvolvimento está atrelado ao empenho de empresas e grupos interessados em desenvolver modelos de negócio baseados na prestação de serviços e suporte técnico, a exemplo do Blender, que alcançou estágio de desenvolvimento compatível com seus equivalentes comerciais. As ferramentas livres, de uma forma geral, não parecem estar na corrida pela liderança estabelecida por produtos comerciais mais avançados. Primeiro, porque a própria filosofia do *software* livre não contempla uma disputa pelo mercado, mas, sim, a constituição de uma alternativa, podendo, inclusive, atuar em conjunto com ferramentas comerciais, disponibilizando versões que "rodam" em sistemas operacionais proprietários. Segundo porque é construído de forma cooperativa, com base em trabalho voluntário, dependendo de um interesse coletivo e de idéias que não estão necessariamente relacionadas com a constituição de um capital, mas na troca de informações que vão adicionando às ferramentas capacidades que a sua própria "arquitetura" permite que sejam desenvolvidas, em prol de necessidades coletivas. Assim, pode-se dizer que esta categoria de programa não "cria necessidades", mas atende a elas de uma forma gradual, enquanto que ferramentas proprietárias passam a agregar cada vez mais possibilidades, influenciando o próprio processo de projeção, e criando novas facilidades constantemente.

A tendência atual é a de integração das ferramentas CAD em torno da completa representação tridimensional do edifício sobre uma complexa base de dados, que carrega informações físicas e funcionais da edificação, dando origem a uma tecnologia emergente denominada de BIM. O atual estágio de desenvolvimento dos programas livres gráficos apresenta uma defasagem muito grande com relação aos avanços alcançados por essa tecnologia e, nesse sentido, dificilmente poderiam lograr a posição de concorrentes aos produtos proprietários, reservada, até o momento, a alternativa de serem utilizados, principalmente na fase de documentação do projeto e para análises de cunho formal/estético da volumetria e composição dos componentes arquitetônicos. As perspectivas de que alcancem o atual estágio de desenvolvimento da tecnologia de automação do projeto são remotas, diante do monopólio estabelecido por determinadas marcas e patentes da indústria do *software*. Apenas seriam plausíveis na medida em que Universidades e Centros de pesquisa, voltados para a área da Arquitetura, Urbanismo e Ciência da Computação, encontrassem a motivação necessária para formar grupos

interdisciplinares dedicados ao desenvolvimento dessas ferramentas e, mesmo assim, esbarrariam em questões relacionadas com o direito da propriedade intelectual, requerendo esforço extra para desenvolver tecnologia que não tivesse dono.

Conforme afirmado na introdução desse trabalho, a prática indiscriminada de distribuição e uso ilegal de *softwares* proprietários de qualquer natureza, para qualquer fim, expõe os usuários a graves penalidades previstas em lei. Embora o viés de desenvolvimento desse estudo não tenha explorado o universo dessas penalidades por questões metodológicas, considerando os objetivos da pesquisa e a área do conhecimento a que se dedica; com base nas observações, comentários e comportamento dos indivíduos que participaram do experimento, conclui-se que, aliado aos fatores acima enumerados, de uma forma geral, configura-se também como barreira à disseminação dos programas livres gráficos a sensação de impunidade, de quase certeza de que não chegarão a sofrer nenhum tipo de penalidade, caso façam uso de programas gráficos proprietários em situação irregular.

Dessa forma, o *software* livre vem ganhando espaço como alternativa de inclusão digital de camadas menos favorecidas da sociedade, notoriamente absorvidos no contexto do papel do Estado em reduzir custos, disseminar o acesso à tecnologia computacional e ao próprio conhecimento. Os impactos que esse fenômeno vem trazendo para diversas da atividade acadêmica, aponta para a necessidade de se trazer o tema para o contexto do ensino da informática aplicada ao projeto, como forma de se prever, e prover, o atendimento às necessidades emergentes da política de implementação do *software* livre adotada pelo Estado, cujos impactos já se sentem em infocentros e laboratórios de informática da rede pública de ensino que passaram a receber computadores baseados em sistemas livres, aos quais a Indústria do *software* CAD proprietário ainda não voltou sua atenção, haja visto a carência de programas gráficos comerciais compatíveis com esses sistemas, a exemplo do Linux.

A reboque dessa demanda, a incorporação de programas livres gráficos ao elenco de ferramentas adotadas nas disciplinas voltadas para a informatização do processo projetual, nas Faculdades de Arquitetura e Urbanismo, pode contribuir para ampliar os conhecimentos dos estudantes acerca de alternativas de automação do projeto e de acesso à tecnologia CAD, importante para aumentar o grau de competência do futuro profissional em julgar qual o melhor caminho a adotar diante da prerrogativa da automação, ao meso tempo em que contribui para o fortalecimento das comunidades em torno do desenvolvimento de *softwares*

gráficos livres, já que o uso da ferramenta para benefício próprio constitui motivação que leva indivíduos a colaborarem, de diversas formas, com projetos de desenvolvimento de programas livres.

As conclusões apresentadas trazem, nas suas entrelinhas, recomendações para o desenvolvimento dos programas gráficos livres selecionados, baseados nos resultados de uma avaliação qualitativa, onde a participação de estudantes e profissionais de arquitetura foi de fundamental importância para a identificação das variáveis que influenciam a qualidade em uso desses programas.

Nesse contexto, recomenda-se o maior envolvimento de arquitetos, engenheiros e outros profissionais da área de arquitetura, engenharia e construção nas comunidade formadas em torno do desenvolvimento dos *softwares* livres gráficos, como forma de colaborar com *feed back* para o aperfeiçoamento dessas ferramentas, visando melhorar a apreensão; incrementar os recursos disponíveis e suas funcionalidades, no sentido de que se possa usufruir, de forma plena, dessa tecnologia no atendimento das necessidade de desenvolvimento e automação do projeto. A aparente ausência de contingentes com este perfil envolvidos com o projeto de desenvolvimento dessas ferramentas, certamente está relacionado à falta de informação e com o fato de se achar que esta participação implica em desenvolver atividades de programação, o que não é verdade, pois, como visto neste estudo, existe uma parcela significativa de indivíduos que contribuem para projetos de *software* livre informando problemas e discutindo as funcionalidades do programa, mas, para isso, é preciso que a filosofia do *software* livre seja disseminada nos meios especializados através do desenvolvimento de trabalhos que fomentem a pesquisa em torno deste tema, ainda pouco explorado. Neste sentido, abre-se campo para o desenvolvimento de novos estudos que explorem o universo de ferramentas distribuídas sob os termos de outros tipos de licença *Open Source*, que não a GNU/GPL, considerada neste trabalho, e que veio ao conhecimento a partir do processo de seleção das ferramentas estudadas. Outro campo se abre com a constatação da existência de ferramentas livres para as mais diversas finalidades, na perspectiva de se investigar a possibilidade de utilização de um pacote de *software* livre para automação não só do projeto, mas de todos os processos que envolvem a atuação de escritórios de arquitetura, bem como a existência de ferramentas dessa natureza que atendam a necessidades oriundas de áreas correlatas da engenharia e da construção civil.



## REFERÊNCIAS

- AMORIM, A. L. . Linguagem, Informação e Representação do Espaço. In: SIGRADI: CONGRESSO IBERO AMERICANO DE GRÁFICA DIGITAL, 4, 2000. **Anais...** Rio de Janeiro: UFRJ/PROURB, 2000. v. I.
- AMORIM, A. L. **Novas Tecnologias no Projeto X Novas Metodologias de Trabalho.** Salvador. Faculdade de Arquitetura da Universidade Federal da Bahia. Conferência apresentada em concurso público para professor titular. 1999, 21 p.
- AMORIM, A. L. **Tecnologias CAD no Ensino de Arquitetura e Engenharia.** São Paulo. Tese (Doutorado) Escola politécnica da USP. 1997, 215p.
- APACHE SOFTWARE FOUNDATION. Subversion. **Homepage na internet...** Disponível em: <<http://subversion.apache.org/>>. Acesso em: 22 jun. 2011.
- APPLE MACINTOSH.JPG 2004. Altura: 1266 pixels. Largura: 1017 pixels. 528.912 bytes. 520 Kb. Formato JPEG Disponível em: <<http://www.aresluna.org/attached/computerhistory/articles/macintoshbytepreview/pics/photo1>>. Acesso em: 22 jun. 2011.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6492:** Representação de Projetos de Arquitetura. Rio de Janeiro. 1994. 27 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1:** Engenharia de Software – Qualidade de produto de software. Parte 1: Modelo de qualidade. Rio de Janeiro. 2003a. 21p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 12207:** Engenharia de Sistemas e Software - Processos de ciclo de vida de software. Rio de Janeiro. 2009. 35 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-1:** Tecnologia de Informação - Avaliação de produto de software. Parte 1: Visão geral. Rio de Janeiro. 2001a. 13 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-2:** Engenharia de Software - Avaliação de produto. Parte 2: Planejamento e gestão. Rio de Janeiro. 2003b. 11 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-4:** Engenharia de Software - Avaliação de produto. Parte 4: Processo para adquirentes. Rio de Janeiro. 2003c. 31 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-5:** Tecnologia de Informação - Avaliação de produto de software. Parte 5: Processo para avaliadores. Rio de Janeiro. 2001b. 27 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-6:** Tecnologia de Informação - Avaliação de produto de software. Parte 6: Documentação de módulos de avaliação. Rio de Janeiro. 2004. 32 p.

AUGUSTO, M. P. **Um Estudo Sobre as Motivações e Orientações de Usuários e Programadores Brasileiros de Software Livre**. Rio de Janeiro. Dissertação (Mestrado) Universidade Federal do Rio de Janeiro, Instituto COPPEAD de Administração. 2003, 111p. Disponível em:

<[http://www2.coppead.ufrj.br/port/index.php?option=com\\_docman&Itemid=204&task=cat\\_view&gid=46&dir=DESC&order=date&limit=10&limitstart=280](http://www2.coppead.ufrj.br/port/index.php?option=com_docman&Itemid=204&task=cat_view&gid=46&dir=DESC&order=date&limit=10&limitstart=280)> Acesso em: 10 jul. 2011.

AUTODESK. AutoCAD Architecture: purpose-built for productivity. Autodesk Inc.

**Homepage na internet...** c2010. Disponível em:

<[http://images.autodesk.com/adsk/files/autocad\\_architecture\\_2012\\_brochure.pdf](http://images.autodesk.com/adsk/files/autocad_architecture_2012_brochure.pdf)> Acesso em: 26 jul. 2011.

AZEVEDO, S. P. **Modelo de Avaliação da Qualidade Funcional de Software**. Novo Hamburgo. Monografia (Graduação). Instituto de Ciências Exatas e Tecnológicas do Centro Universitário Feevale. 2008. 33p.

BARR E. R., JURICIC D. From Drafting to Modern Design Representation: The evolution of Engineering Design Graphics. **Journal of Engineering Education**. Washington. v. 83, n. 2, p. 263-270, jul. 1994. Disponível em: <<http://www.jee.org/1994/july/124.pdf>> Acesso em: 26 jul. 2011.

BESANT, C. B. **CAD/CAM: Projeto e Fabricação com o Auxílio de Computador**. Tradução Ricardo Reinprecht. 2. ed. Rio de Janeiro: Campus, 1985. 249p.

BLENDER FOUNDATION. Blender Art Gallery. **Homepage na internet...** Disponível em: <<http://www.blender.org/features-gallery/gallery/art-gallery/>>. Acesso em: 03 jul. 2011a.

BLENDER FOUNDATION. Blender Foundation History. **Homepage na internet...** Disponível em: <<http://www.blender.org/blenderorg/blender-foundation/history/>>. Acesso em: 03 jul. 2011b.

BLENDER FOUNDATION. Blender User Community. **Homepage na internet...** Disponível em: <<http://www.blender.org/community/user-community/>>. Acesso em: 03 jul. 2011c.

BLENDER FOUNDATION. Índice de versões do Blender. **Homepage na internet...** Disponível em: <<http://download.blender.org/release/>>. Acesso em: 21 jul. 2011d.

BLENDER FOUNDATION. Statistics Analysis 04-05. **Homepage na internet...** Disponível em: <<http://www.blender.org/bf/stats0405.htm>>. Acesso em: 01 jul. 2011e.

BLENDER FOUNDATION. Website Statistics. **Homepage na internet...** Disponível em: <<http://www.blender.org/blenderorg/blender-foundation/website-statistics/>>. Acesso em: 01 jul. 2011f.

BORGES, M. M. **A Projeção e as Formas de Representação do Projeto**. Rio de Janeiro. Dissertação (Mestrado). Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro. 1998. 130p.

BRASIL. Ministério da Ciência e Tecnologia. **Chamada Pública MCT/FINEP/CT-INFO 01/2003 Software Livre**. 2003. Disponível em: <[http://www.finep.gov.br/fundos\\_setoriais/ct\\_info/resultados/resultado\\_chamada\\_publicaMCT\\_Finep\\_ct\\_info\\_01\\_2003.pdf](http://www.finep.gov.br/fundos_setoriais/ct_info/resultados/resultado_chamada_publicaMCT_Finep_ct_info_01_2003.pdf)>. Acesso em: 25 jul. 2011

BRASIL. Ministério da Ciência e Tecnologia. **Chamada Pública MCT/FINEP/CT- INFO - Grade - 01/2004**. 2004. Disponível em: <[http://www.finep.gov.br/fundos\\_setoriais/ct\\_info/resultados/resultado\\_CT\\_INFO\\_Grade\\_01\\_2004..PDF](http://www.finep.gov.br/fundos_setoriais/ct_info/resultados/resultado_CT_INFO_Grade_01_2004..PDF)>. Acesso em: 25 jul. 2011.

BRASIL. Ministério da Fazenda: Serviço Federal de Processamento de Dados (SERPRO). [200-] **Homepage na internet...** Disponível em: <<http://www.serpro.gov.br/tecnologia/software-livre>>. Acesso em: 25 jul. 2011.

CARDOSO, F. F. **A Utilização de Computadores no Projeto do Edifício: conceitos e perspectivas**. São Paulo. Dissertação (Mestrado). Escola Politécnica da Universidade de São Paulo. 1986. 345p.

CONISLI. Congresso Internacional de Software Livre. 8., São Paulo, SP. **Homepage na internet...** Disponível em: <<http://www.conisli.org.br/>> Acesso em: 26 jul. 2011.

CORDEIRO, A. L. M. **O Projeto do Edifício em Ambientes Informatizados: Uma abordagem macroergonômica**. João Pessoa. Dissertação (Doutorado). Universidade Federal da Paraíba. 2007 170 p.

CORSEUIL, E. T. L. et al. **Buscando o Uso Operacional de Realidade Virtual em Grandes Modelos de Engenharia**. In: SYMPOSIUM ON VIRTUAL REALITY, 6. Ribeirão Preto. 2003. p. 187 – 98. Disponível em: <[http://web.tecgraf.puc-rio.br/publications/artigo\\_2003\\_uso\\_operacional\\_realidade\\_virtual.pdf](http://web.tecgraf.puc-rio.br/publications/artigo_2003_uso_operacional_realidade_virtual.pdf)> Acesso em: 26 jul. 2011.

EASTMAN. C. M. **Building Product Model: Computer Enviroments Supporting Design and Construction**. Florida: CRC Press LCC. 1999. 403 p.

ENSL. Encontro Nordestino de Software Livre e I Encontro Alagoano de Software Livre. 5., 2011, Maceió. **Homepage na internet...** Disponível em: <<http://ensl.org.br/>>. Acesso em: 26 jul. 2011.

FALCÃO, J. et al. **Estudo Sobre o Software Livre: Comissionado pelo Instituto Nacional de Tecnologia (ITI)**. Rio de Janeiro: Fundação Getúlio Vargas, 2005. 121 p. Disponível em: <[http://www.softwarelivre.gov.br/publicacoes/Estudo\\_FGV.pdf](http://www.softwarelivre.gov.br/publicacoes/Estudo_FGV.pdf)>. Acesso em: 26 jul. 2011.

FASOL. Fórum Amazônico de Software Livre. 3., 2011, Santarém, PA. **Homepage na internet...** Disponível em: <<http://www.fasol.org.br/>>. Acesso em: 26 jul. 2011.

FELLER, J. et al. **Perspectives on Free and Open Source Software**. Londres: MIT Press. 2005. 538 p.

FERREIRA, F. S. **Ferramenta para Avaliação da Qualidade de Produtos de Software Baseado nas Normas ISO/IEC 9126 e NBR ISO/IEC 12119**. Blumenau. Monografia (Graduação). Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau. 2003.103p.

FERREIRA, R. F.; CALVOSO, G. G.; GONZALES, C. B. L. Caminhos da Pesquisa e a Contemporaneidade. *Psicologia: Reflexão e Crítica*. Porto Alegre, v.15, n. 2, p. 243-250, 2002. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0102-797220022000200002](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-797220022000200002)> Acesso em: 19 jul. 2011.

FISL. Fórum Internacional de Software Livre. 12., 2011, Porto Alegre, RS. **Homepage na internet...** Disponível em: <<http://softwarelivre.org/fisl12/o-evento>>. Acesso em: 26 jul. 2011.

FREE SOFTWARE FOUNDATION. O que é Copyleft? c1996b. **Homepage na internet...** Disponível em: <<http://ramonflores.br.tripod.com/GNU/copyleft.html>>. Acesso em: 10 jul. 2011.

FREE SOFTWARE FOUNDATION. O que é o Software Livre? c1996a. **Homepage na internet...** Disponível em: <<http://www.gnu.org/philosophy/free-sw.pt-br.html>>. Acesso em: 10 jul. 2011.

FREE SOFTWARE FOUNDATION. Vender Software livre. c1996c. **Homepage na internet...** Disponível em: <<http://ramonflores.br.tripod.com/GNU/vendendo.html>>. Acesso em: 10 jul. 2011.

FSLDC. Fórum de Software Livre de Duque de Caxias. 3., 2011, Duque de Caxias, RJ. **Homepage na internet...** Disponível em: <<http://forumsoftwarelivre.com.br/2011/>>. Acesso em: 26 jul. 2011.

FTSL. Fórum de Tecnologia em Software Livre. 4., 2011, Curitiba, PR. **Homepage na internet...** Disponível em: <<http://www.ftsl.org.br/o-evento>>. Acesso em: 26 jul. 2011.

HEALY, K.; SCHUSSMAN, A.: **The Ecology of Open-Source Software Development**. Arizona. Artigo. Department of Sociology, University of Arizona. 2003. 24 p. Disponível em: <[http://scholar.google.com.br/scholar?q=HEALY,+K.%3B+SCHUSSMAN,+A.:+The+Ecology+of+Open-Source+Software+Development&hl=pt-BR&as\\_sdt=0&as\\_vis=1&oi=scholar](http://scholar.google.com.br/scholar?q=HEALY,+K.%3B+SCHUSSMAN,+A.:+The+Ecology+of+Open-Source+Software+Development&hl=pt-BR&as_sdt=0&as_vis=1&oi=scholar)> Acesso em: 26 jul. 2011.

IBM\_PC\_5150.JPG 2011. Altura: 740 pixels. Largura:1024 pixels. 332.531 bytes. 316 Kb. Formato JPEG Disponível em: <<http://escreveassim.com.br/2011/06/16/100-anos-de-ibm/>> Acesso em: 26 jul. 2011.

INKSCAPE BRASIL. Comunidade Brasileira do Projeto Inkscape.org. **Homepage na internet...** Disponível em: <<http://wiki.softwarelivre.org/InkscapeBrasil/WebHome>>. Acesso em: 03 jul. 2011.

INKSCAPE.ORG Open Source Scalable Vector Graphics Editor. **Homepage na internet...** Disponível em: <<http://inkscape.org/>>. Acesso em: 03 jul. 2011.

INSTITUTO DE TECNOLOGIA DA INFORMAÇÃO. Casa Civil da Presidência da República Federativa do Brasil. Software Livre no Governo do Brasil. [200-] **Homepage na internet...** Disponível em: <<http://www.iti.gov.br/twiki/bin/view/Swlivre/WebHome>>. Acesso em: 26 jul. 2011.

KALISPERIS, N. L. Computer Based Architectural Design Representation. In: NUTAU 1996: SEMINÁRIO INTERNACIONAL NÚCLEO DE PESQUISA EM TECNOLOGIA DA ARQUITETURA E URBANISMO. São Paulo. **Anais...** São Paulo.: FAU/USP, 1997. p. 219-231.

KASBURG, A. **Avaliação de Qualidade de Software de Gestão Integrada Utilizando as Normas ISO/IEC 9126 e ISO/IEC 14598-1**. Blumenau. Monografia (Graduação). Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau. 2001. 84p.

KRISHNAMURTHY, S. Cave or Community? An Empirical Examination of 100 Mature Open Source Projects. **First Monday**: Jornal online da Universidade de Illinois. Chicago. v. 7, n. 6, jun. 2002. Não paginado. Disponível em: <<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960/881>>. Acesso em: 26 jul. 2011.

KOFFICE. Koffice. **Homepage na internet...** Disponível em: <<http://sourceforge.net/>> Acesso em 29. jul. 2011.

KRYGIEL, E; NIES, B.: **Green BIM: Successful Sustainable Design with Building Information Model**. Indianápolis: Sybex. 1. ed. 2008. 241p.

K3D.ORG K3d. **Homepage na internet...** Disponível em: <<http://www.k-3d.org/>>. Acesso em: 03 jul. 2011.

LATINOWARE. Conferência Latino-Americana de Software Livre. 8., 2011, Foz do Iguaçu, PR. **Homepage na internet...** Disponível em: <<http://www.latinoware.org/>>. Acesso em: 26 jul. 2011.

LAURENT, A. M. **Open Source & Free Software Licensing**. 1. ed. Sebastopol: O'reilly. 2004. 193 p.

LAWSON, B. Cad na arquitetura: a história até agora. **Graf& Tec**: revista da Associação Brasileira de Expressão Gráfica (ABEG), São Paulo, SP. n.6, 2 sem.1999, p. 31-59.

LEIGHTON, N. L. **El Ordenador en el Estudio del Arquitecto**: guia práctica y aplicaciones. Barcelona: Gustavo Gilli. 1. de. 1987. 186 p.

LEVY, S. **Hackers: Heroes of Computer Revolution**. Nova York: Delta Book. 1994. 367 p. Disponível em: <[http://housem4dz.ifrance.com/book/hackers\\_-\\_heroes\\_of\\_the\\_computer\\_revolution\\_-\\_steven\\_levy.pdf](http://housem4dz.ifrance.com/book/hackers_-_heroes_of_the_computer_revolution_-_steven_levy.pdf)>. Acesso em: 26 jul. 2011.

LIBRECAD. LibreCAD, Open Source 2D CAD. **Homepage na internet...** Disponível em: <<http://librecad.org/cms/home.html>>. Acesso em: 21 nov. 2011.

LISA 1 - BROSHURE.JPG [200-] Altura: 454 pixels. Largura: 450 pixels. 65.754 bytes. 68 Kb. Formato JPEG Disponível em: <<http://www.mauseum.net/Lisa/>>. Acesso em: 22 jun. 2011.

MANDEL, A.; SIMON, I.; LYRA, J. L. Informação: Computação e Comunicação. **Revistausp**. São Paulo. n. 35, 1997. Não paginado. Disponível em: <<http://www.ime.usp.br/~is/abc/abc/abc.html>> Acesso em: 21 de jul. 2011.

MARIA, J. S. **Roteiro Para Elaboração de Métricas no Processo de Desenvolvimento de Software Utilizando Métodos DAMAIC e GQM**. Novo Hamburgo. Monografia (Graduação). Universidade Feevale. 2010. 77 p.

MARIANI, A. C. O Mundo dos Atores: uma perspectiva de introdução à programação orientada a objetos. **Revista Brasileira de Informática e Educação**: Revista online da Sociedade Brasileira de Computação (SBC). Porto Alegre. v. 5. 1999. Não paginado. Disponível em: <<http://bibliotecadigital.sbc.org.br/download.php?paper=889>> Acesso em: 26 jul. 2011.

MASSEY, B. Where Do Open Source Requirements Come From (And What Should We Do About It)? In: Proceedings of the 2nd Workshop On Open-Source Software Engineering. **Anais eletrônicos...** Orlando, FL. 2002. Não paginado. Disponível em: <<http://opensource.ucc.ie/icse2002/Massey.pdf>>. Acesso em: 10 jul. 2011.

MATTHEWS, D. Daniel's K3D Development Blog. **Homepage na internet...** Disponível em: <<http://k-3d.org/blogs/dsmatthews/>>. Acesso em: 03 jul. 2011.

MAZZA, S. C. **CAD na Arquitetura e Engenharia**. Salvador. Monografia (Especialização). Faculdade de Arquitetura da Universidade Federal da Bahia. 2000. 75p.

MITCHEL, W. J. **A Lógica da Arquitetura**: projeto, composição e cognição. Tradução Gabriela Celani. Campinas: Unicamp. 2008. 304 p.

MYINT, S.; BOONYAWAT, T.; JACKRAPONG, A. Development of Internet CAD System for CE Enviroment. **Thammasat International Journal of Science and Technology**. Tailândia. v. 4, n. 2. jul. 1999. Não paginado. Disponível em: <[http://www.tijsat.tu.ac.th/issues1999/no2/1999\\_V4\\_No2\\_7.PDF](http://www.tijsat.tu.ac.th/issues1999/no2/1999_V4_No2_7.PDF)> Acesso em: 26 jul. 2011.

NAKAKOJI et al. Evolution Patterns of Open-Source Software Systems and Communities. In: Proceedings of the International Workshop on Principles of Software Evolution. **Anais eletrônicos...** 2002, Orlando, FL. Não paginado. Disponível em: <<http://portal.acm.org/citation.cfm?id=512055>> Acesso em: 10 jul. 2011.

NAVEIRO, R. M.; BORGES, M. M. Projetação e Formas de Representação do Projeto. In: GRAPHICA 98: SIMPÓSIO NACIONAL DE GEOMETRIA DESCRITIVA E DESENHO TÉCNICO, 13. 1998, Feira de Santana **Anais...** Feira de Santana: UEFS, 1998. p 51 – 61.

NEVES, J. L. Pesquisa Qualitativa – características, usos e possibilidades. **Caderno de Pesquisa em Administração**, FEA-USP, São Paulo, v. 1, n. 3, p. 1-5, 2º sem. 1996. Não paginado. Disponível em: <<http://www.ead.fea.usp.br/cad-pesq/arquivos/C03-art06.pdf>> Acesso em: 19 jul. 2011.

NUNES, R. C. P. **Implementação e Padronização de Sistemas CAD**: uma análise dos escritórios do Rio de Janeiro. Niterói. Dissertação (Mestrado) Universidade Federal Fluminense. 1997, 180 p.

PIRES, F. P. O Surgimento dos Primeiros Computadores. **Educação Pública**: Publicação Seriada da Fundação Centro de Ciência e Educação Superior a Distância do Estado do Rio de Janeiro, Rio de Janeiro. [200-] ISSN 1984-629. Disponível em: <<http://www.educacaopublica.rj.gov.br/biblioteca/geografia/0000.htm>> Acesso em 26 jul. 2011.

PROJETO SOFTWARE LIVRE BAHIA. Projeto Software Livre Bahia. c2003. **Homepage na internet...** Disponível em: <<http://psl-ba.softwarelivre.org/>>. Acesso em: 26 jul. 2011.

REGO, R. M. **Arquitetura e Tecnologias Computacionais**: novos instrumentos mediadores e as possibilidades de mudança no projeto projetual. Salvador. Dissertação (Mestrado). Faculdade de Arquitetura da Universidade Federal da Bahia. 2000. 173p.

REIS, C. R.; **Caracterização de um Processo de Software para Projetos de Software Livre**. São Carlos. Dissertação (Mestrado). Instituto de Ciências Matemáticas e da Computação da Universidade de São Paulo. 2003, 247p. Disponível em: <[http://async.com.br/~kiko/dissert\\_usp.pdf](http://async.com.br/~kiko/dissert_usp.pdf)> Acesso em: 30 ago. 2006.

RIBBONSOFT. Qcad Documentation. **Homepage na internet...** Disponível em: <[http://www.qcad.org/qcad\\_doc.html](http://www.qcad.org/qcad_doc.html)>. Acesso em: 03 jul. 2011a.

RIBBONSOFT. Qcad Product Overview. **Homepage na internet...** Disponível em: <<http://www.qcad.org/>>. Acesso em: 03 jul. 2011b.

RINCON, A. M. Qualidade de Software. In: ENCONTRO DE ESTUDANTES DE INFORMÁTICA DO TOCANTINS. Palmas. 2009. **Anais...** Palmas: Centro Universitário Luterano de Palmas. 2009. p. 75-86.

ROMEIRO, E. **CAD na Indústria Implantação e Gerenciamento**. Rio de Janeiro: UFRJ. v.1. 1. ed. 1997, 176p.

RORATTO, J. M. Posições Subjetivistas e Objetivistas de Ciência: a hermenêutica como fundamento da pesquisa qualitativa. **Roteiro**: Santa Catarina. v35, n.1, 2010.p. 175 -192. Disponível em: <<http://editora.unoesc.edu.br/index.php/roteiro/article/view/233>> Acesso em: 21 jul. 2011.

SERRA, G. G. Teoria, Simulação e Modelo Na Arquitetura. In: NUTAU 1995: ENCONTRO NACIONAL DE MODELOS DE "SIMULAÇÃO DE AMBIENTES". São Paulo. 1995. **Anais...** São Paulo: FAU-USP, 1995. p. 09 - 18.

SOARES, C. C. P. Desenho, Representação e os Processos de Produção. In: ENCONTRO REGIONAL DE EXPRESSÃO GRÁFICA, 5. Salvador. 2006. **Anais eletrônicos...** Salvador. PPGAU/FAUFBA, 2006. 1 CD-ROM

SOCIEDADE SOFTEX. **O Impacto do Software Livre e de Código Aberto na Indústria de Software do Brasil**. Campinas, SP. Softex, 2005. 76 p. Disponível em: <[http://www.softex.br/portal/softexweb/uploadDocuments/\\_observatorio/SWL%20MCT%20portugues%20site.pdf](http://www.softex.br/portal/softexweb/uploadDocuments/_observatorio/SWL%20MCT%20portugues%20site.pdf)> Acesso em: 26 jul. 2011.

SOCIEDADE SOFTEX. Sociedade SOFTEX. [200-] **Homepage na internet...** Disponível em: <[http://www.softex.br/\\_asoftex/sociedadeSoftex.asp](http://www.softex.br/_asoftex/sociedadeSoftex.asp)>. Acesso em: 26 jul. 2011.  
SODRÉ, C. C. P. **Norma ISO/IEC 9126: Avaliação de Qualidade de Produto de Software**. Londrina. Monografia (Graduação). Departamento de Computação da Universidade Estadual de Londrina. 2006.53 p.

SOURCEFORGE. Sourceforge.net. c2011. **Homepage na Internet...** Disponível em: <<http://sourceforge.net>> Acesso em: 29 jul. 2011.

STALLMAN, R. Free Software Foundation. **O Manifesto GNU**. 1985. Não paginado. Disponível em: <<http://www.gnu.org/gnu/manifesto.pt-br.html>>. Acesso em: 10 jul. 2011.

STALLMAN, R. Free Software Foundation. **Porque o Software Deveria Ser Livre**. 1992. Não paginado. Disponível em: <<http://www.gnu.org/philosophy/shouldbefree.pt-br.html>>. Acesso em: 10 jul. 2011.

STALLMAN, R. **The GNU Project**. Free Software Foundation. c1998. Não paginado. Disponível em: <<http://www.gnu.org/gnu/thegnuproject.html>> Acesso em: 26 jul. 2011.

STEFANUTO, G.N.; SALLES, S.F. O Impacto Software Livre e de Código Aberto (SL/CA) nas Condições de Apropriabilidade na Indústria de Software Brasileira. In: SEMINÁRIO LATINO-IBEROAMERICANO DE GESTIÓN TECNOLÓGICA. 11. Salvador. **Anais eletrônicos...** Salvador: ALTEC, 2005. Disponível em: <[http://www.softex.br/portal/softexweb/uploadDocuments/\\_observatorio/altec\\_apropriabilidade\\_sl.pdf](http://www.softex.br/portal/softexweb/uploadDocuments/_observatorio/altec_apropriabilidade_sl.pdf)> Acesso em: 10 jul. 2011.

SUTHERLAND, I. E. **Sketchpad: A man-machine graphical communication system**. Cambridge: University Of Cambridge Computer Laboratory, 2003. 185 p. Disponível em: <<http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf>>. Acesso em: 10 jul. 2011.

SYNFIG BR. Comunidade Brasileira de Usuários do Synfig. **Homepage na internet...** Disponível em: <<http://synfig-br.org/>>. Acesso em: 03 jul. 2011.

SYNFIG STUDIO. Synfig Studio: Open Source Animation Software. **Homepage na internet...** Disponível em: <<http://synfig.org/>>. Acesso em: 03 jul. 2011.

TELES, F. S. **Um Processo para Análise de desempenho de Produto de Software**. Recife. Monografia (Graduação). Centro de Ciências da Computação da Universidade Federal de Pernambuco. 2005. 77 p. Disponível em: <<http://www.cin.ufpe.br/~tg/2004-2/fst.pdf>>. Acesso em: 21 jul. 2011.

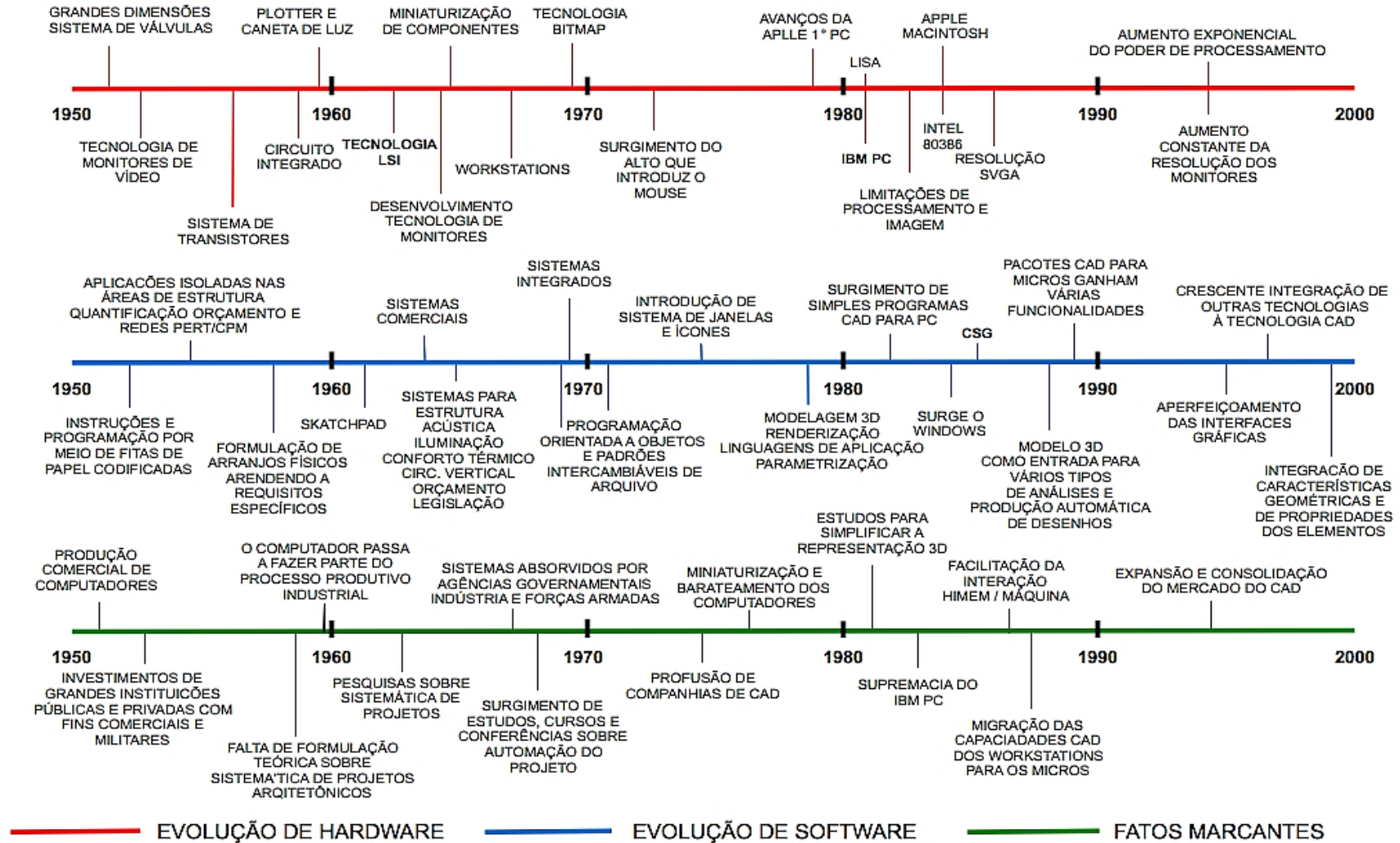
WILSON, G. *Is the Open-Source Community Setting a Bad Example?* **IEEE Software**: Los Angeles. v. 16, n. 1. 1999. p. 23–25. Disponível em: <<http://www.raminsoftworx.com/elec443/lectures/open-source-5.pdf>> Acesso em: 26 jul. 2011.



ZENG, J., CHEN, W., DING, Q. *A Web-based CAD system*. Elsevier: Rio de Janeiro. **Journal of Materials Processing Technology**. v. 139. n. 1. 2003. p. 229 - 232. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0924013603002255>> Acesso em 26 jul. 2011.

## APÊNDICE A

### EVOLUÇÃO HISTÓRICA DA CONSOLIDAÇÃO DO USO DO COMPUTADOR NO CAMPO DO PROJETO (1950/2000)



Fonte: Elaborado pelo autor

## **APÊNDICE B**

**ATRIBUTOS DAS QUALIDADES INTERNA E EXTERNA: NBR ISO/IEC 9126-1**

## APÊNDICE B

### ATRIBUTOS DAS QUALIDADES INTERNA E EXTERNA: NBR ISO/IEC 9126-1

#### Quadro I – Definições dos atributos de qualidade relacionados com a Funcionalidade

MODELO DE QUALIDADE EXTERNA E INTERNA DO PRODUTO DE SOFTWARE	
CARACTERÍSTICA	SUBCARACTERÍSTICA
<b>Funcionalidade:</b> Capacidade de prover funções que atendam às necessidades explícitas e implícitas, quando o software estiver sendo utilizado em condições específicas.	Adequação
	Acurácia
	Interoperabilidade
	Segurança de acesso
	Conformidade

Fonte: Elaborado pelo autor com base em (ABNT, 2003a).

#### Quadro II – Definições dos atributos de qualidade relacionados com a Confiabilidade

MODELO DE QUALIDADE EXTERNA E INTERNA DO PRODUTO DE SOFTWARE	
CARACTERÍSTICA	SUBCARACTERÍSTICA
<b>Confiabilidade:</b> Capacidade de manter um nível de desempenho especificado, quando usado em condições específicas.	Maturidade
	Tolerância a falhas
	Recuperabilidade
	Conformidade

Fonte: Elaborado pelo autor com base em (ABNT, 2003a).

#### Quadro III – Definições dos atributos de qualidade relacionados com a Usabilidade

MODELO DE QUALIDADE EXTERNA E INTERNA DO PRODUTO DE SOFTWARE	
CARACTERÍSTICA	SUBCARACTERÍSTICA
<b>Usabilidade:</b> Capacidade de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.	Apreensibilidade
	Operacionalidade
	Atratividade
	Conformidade

Fonte: Elaborado pelo autor com base em (ABNT, 2003a).

(CONTINUA)

## APÊNDICE B

### ATRIBUTOS DAS QUALIDADES INTERNA E EXTERNA: NBR ISO/IEC 9126-1

(CONTINUAÇÃO)

**Quadro IV** – Definições dos atributos de qualidade relacionados com a Eficiência

MODELO DE QUALIDADE EXTERNA E INTERNA DO PRODUTO DE SOFTWARE	
CARACTERÍSTICA	SUBCARACTERÍSTICA
<b>Eficiência:</b> Capacidade de apresentar desempenho apropriado, relativo à quantidade de recursos utilizados, sob condições especificadas.	Comportamento em relação ao tempo
	Utilização de recursos
	Conformidade

Fonte: Elaborado pelo autor com base em (ABNT, 2003a).

**Quadro V** – Definições dos atributos de qualidade relacionados com a Manutenibilidade

MODELO DE QUALIDADE EXTERNA E INTERNA DO PRODUTO DE SOFTWARE	
CARACTERÍSTICA	SUBCARACTERÍSTICA
<b>Manutenibilidade:</b> Capacidade de ser modificado, incluindo correções, melhorias ou adaptações devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.	Estabilidade
	Testabilidade
	Conformidade

Fonte: Elaborado pelo autor com base em (ABNT, 2003a).

**Quadro VI** – Definições dos atributos de qualidade relacionados com a Portabilidade

MODELO DE QUALIDADE EXTERNA E INTERNA DO PRODUTO DE SOFTWARE	
CARACTERÍSTICA	SUBCARACTERÍSTICA
<b>Portabilidade:</b> Capacidade de ser transferido de um ambiente para outro (organizacional, de hardware ou de software).	Adaptabilidade
	Capacidade de ser instalado
	Coexistência
	Capacidade para substituir
	Conformidade

Fonte: Elaborado pelo autor com base em (ABNT, 2003a)

**APÊNDICE C**  
**ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO**

## APÊNDICE C

### CADUNTUN: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 1 – Elaboração da Planta Baixa

Tempo Estimado = 02h e 30min

AVALIAÇÃO DE SOFTWARE: CADuntu						
ROTEIRO PARA ELABORAÇÃO DA PLANTA BAIXA						
ORDEM	TAREFA	MEDIDAS / LOCAÇÃO	LAYER	COR	ESPESSURA	OBSERVAÇÃO
1	Criar Layers Solicitados					Ver TABELA 1.1
2	Desenhar Eixos	Ver Plotagem	Eixo	Bylayer	0.13	
3	Desenhar Parede	Ver Plotagem	Parede	Bylayer	0.50	Espessura da parede: 0.13m
4	Desenhar Aduela	(0.05 x 0.13)m	Esquadria	Cyan	Bylayer	
5	Criar Bloco <b>Aduela1</b>	Existente	Esquadria	Existente	Existente	
6	Criar Bloco Pilar	(0.70 x 0.70)m	Estrutura	Bylayer	Bylayer	Arestas horizontal e vertical medem 0.13m
7	Criar Bloco <b>Pilar</b>	Existente	Estrutura	Bylayer	Bylayer	1- Não hachurar o Pilar antes da inserção do Bloco 2- Escolher ponto central do pilar como ponto de referência
8	Inserir Pilares nos locais indicados	Ver Plotagem	Estrutura	Bylayer	Bylayer	
9	Inserir Hachura dos Pilares	Escala 0.03	Hatch	Bylayer	Bylayer	Estilo de Hachura ANSI 31
10	Inserir Bloco <b>Aduela 1</b>	Ver Plotagem	Esquadria	Cyan	Bylayer	Aduela de posição vertical, próxima da Parede, deve estar afastada 0.015 m da linha de eixo
11	Inserir Bloco Aduela 2	Ver Plotagem	Esquadria	Cyan	Bylayer	
12	Desenhar Porta Interna	(Vão x 0.03)m	Esquadria	Porta = Bylayer Arco = Red	Porta = 0.25 Arco = 0.13	1 - Medir o vão para obter o comprimento da porta 2- Criar o desenho da Porta no local de sua inserção
13	Criar Bloco <b>Porta 60</b>	Existente	Esquadria	Existente	Existente	O Bloco será utilizado caso seja perdido, ou danificado, o desenho da Porta
14	Desenhar linhas da soleira	Ver Plotagem	Piso	Bylayer	Bylayer	Não sobrepor as linhas da soleira nas esquadrias
15	Fazer o espelhamento da Porta 60	Ver Plotagem	Esquadria	Existente	Existente	
16	Inserir Bloco <b>Porta 70</b>	Ver Plotagem	Esquadria	Existente	Existente	
17	Fazer o espelhamento da Porta 70	Ver Plotagem	Esquadria	Existente	Existente	
18	Inserir Bloco <b>Janela PB</b>	Ver Plotagem	Esquadria	Bylayer	Bylayer	
19	Desenhar linhas de piso	Ver Plotagem	Piso	Bylayer	Bylayer	
20	Desenhar Bancada Central	(2.10 x 0.55)m	Layout	Bylayer	Bylayer	
21	Inserir Bloco <b>Bacia</b>	Ver Plotagem	Layout	Bylayer	Bylayer	
22	Inserir Bloco <b>Lavatório</b>	Ver Plotagem	Layout	Bylayer	Bylayer	
23	Verificar Configuração dos Parâmetros de Dimensionamento					Ver TABELA 1.2
24	Congelar Layer Eixo					
25	Desenhar Projeção da Cobertura	Ver Plotagem	Proj	Bylayer	Bylayer	
26	Cotar o Desenho	Ver Plotagem	Cota	Bylayer	Bylayer	
27	Ajustar locação do texto das cotas de espessura das paredes	Ver Plotagem	Cota	Bylayer	Bylayer	É necessário explodir a cota para mover o texto
28	Inserir Bloco <b>Nível PB</b>	Ver Plotagem	Simbolos	Yellow	0.18	
29	Inserir Bloco <b>Corte</b>	Ver Plotagem	Simbolos	Bylayer	Bylayer	
30	Calcular área dos ambientes					Anotar as áreas para posterior inserção
31	Inserir Textos	Ver Plotagem	Texto	Bylayer	Bylayer	Verificar Configurações na TABELA 1.3

Fonte: Elaborado pelo autor.

## APÊNDICE C

### CADUNTUN: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 2 – Elaboração da Fachada Frontal

**Tempo Estimado = 20min**

PROCEDIMENTO PARA CONSTRUÇÃO DA FACHADA FRONTAL						
ORDEM	TAREFA	MEDIDAS / LOCACÃO	LAYER	COR	ESPESSURA	OBSERVAÇÃO
1	Fazer cópia completa da Planta Baixa	Ao critério do usuário				Deve ser preservado o desenho original da Planta Baixa
2	Rrotacionar a Cópia da Planta Baixa	- 90°				Selecionar todo o desenho de uma vez no modo Select Window
3	Fazer linhas de chamada	Ao critério do usuário	0	Bylayer	Bylayer	
4	Fazer linha de Terra	Ao critério do usuário	Piso	Blue	0.60	Deixar espaço suficiente, entre a linha de terra e a planta baixa, para desenhar a elevação
5	Fazer linhas de Referência paralelas à Linha de Terra	Distance = 0.05m Distance = 2.35m Distance = 2.75m Distance = 2.85m	0	Bylayer	Bylayer	Todas as "Distâncias" dadas são em relação à Linha de Terra
6	Inserir Bloco <b>Cobertura</b>	Ver Plotagem	Aresta	Green	0.50	
7	Espelhar o desenho da vista frontal da cobertura	Ver Plotagem	Aresta	Green	0.50	
8	Desenhar vista do Pilar	Ver Plotagem	Aresta	Yellow	0.18	
9	Desenhar vista das Aduelas 1 e 2, e montante superior (lado esquerdo)	Espessura do montante superior = 0.10	Aresta	Bylayer	Bylayer	
10	Desenhar linha do piso	0.05m acima da Linha de Terra	Aresta	Bylayer	Bylayer	
11	Inserir Bloco <b>Esquadria 1 Vista</b>	Locar na interseção interna da linha de piso com a Aduela 2	Aresta	Existente	Existente	
12	Espelhar o desenho da vista frontal das Aduelas, piso e Esquadria 1 Vista	Ver Plotagem	Aresta			As entidades espelhadas assumirão as características das originais

Fonte: Elaborado pelo autor.



## APÊNDICE C

### CADUNTUN: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 3 – Elaboração do Corte A-A

**Tempo Estimado = 40min**

PROCEDIMENTO PARA CONSTRUÇÃO DO CORTE A-A						
ORDEM	TAREFA	MEDIDAS / LOCACÃO	LAYER	COR	ESPESSURA	OBSERVAÇÃO
13	Fazer cópia completa da Fachada Frontal	Ao critério do usuário				Deve ser preservado o desenho original da Fachada Frontal
14	Construir corte da Cobertura	Ver Plotagem	Linhas do contorno da Secção = Estrutura	Bylayer	Bylayer	Para aproveitar o desenho original da vista da cobertura é preciso explodi-lo
			Linhas vistas = Aresta	Bylayer	Bylayer	
15	Inserir hachura na área seccionada da cobertura	Escala 0.03	Hatch	Bylayer	Bylayer	Estilo de Hachura ANSI 31
16	Construir secção do montante superior	(0.13 x 0.10)m	Esquadria	Cyan	0.35	
17	Trocar atributos das linhas correspondentes à vista da Aduela 2	Existente	Aresta	Yellow	Bylayer	
18	Inserir Bloco <b>Janela CT</b>	Ver Plotagem	Esquadria	Bylayer	Bylayer	
19	Representar Piso seccionado	Ver Plotagem	Piso	Blue	0.50	
20	Representar secção da Bancada	Secção (0.55 x 0.07)m	Aresta	Blue	0.50	
21			Vista da Base	Aresta	Yellow	0.25
22	Inserir Bloco Nivel EL	Ver Plotagem	Simbolos	Yellow	0.18	
23	Cotar o Corte	Ver Plotagem	Cota	Bylayer	Bylayer	
24	Ajustar locação do texto da altura do topo do Pilar	Ver Plotagem	Cota	Bylayer	Bylayer	É necessário explodir a cota para mover o texto

Fonte: Elaborado pelo autor.

## APÊNDICE C

### CADUNTUN: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 4 – Elaboração da Fachada Lateral

**Tempo Estimado = 20 min**

PROCEDIMENTO PARA CONSTRUÇÃO DA FACHADA LATERAL						
ORDEM	TAREFA	MEDIDAS / LOCAÇÃO	LAYER	COR	ESPESSURA	OBSERVAÇÃO
25	Rrotacionar a Cópia da Planta Baixa	90°				Selecionar todo o desenho de uma vez no modo Select Window
26	Fazer linhas de chamada	Ao critério do usuário	0	Bylayer	Bylayer	
27	Fazer linha de Terra	Ao critério do usuário	Piso	Blue	0.60	Deixar espaço suficiente, entre a linha de terra e a planta baixa, para desenhar a elevação
28	Fazer linhas de Referência paralelas à Linha de Terra	Distance = 0.05m Distance = 2.35m Distance = 2.75m Distance = 2.85m	0	Bylayer	Bylayer	Todas as "Distâncias" dadas são em relação à Linha de Terra
29	Desenhar a vista lateral da Cobertura	Ver Plotagem	Aresta	Green e Yellow Ver plotagem	Green = 0.35 Yellow = 0.25	
30	Desenhar vista das arestas da parede	Ver Plotagem	Aresta	Yellow	0.25	
31	Desenhar vista do Pilar	Ver Plotagem	Aresta	Bylayer	Bylayer	
32	Desenhar vista das Aduelas 1 e 2, e montante superior	Espessura do montante superior = 0.10	Aresta	Yellow	0.25	
33	Desenhar linha do piso	0.05m acima da Linha de Terra	Aresta	Yellow	0.25	
34	Inserir Bloco <b>Esquadria 2 Vista</b>	Locar na interseção interna da linha de piso com a Aduela 2	Aresta	Existente	Existente	
35	Copiar o desenho da esquadria para as demais locações	Ver Plotagem	Aresta	Existente	Existente	As entidades copiadas assumirão as características da original
36	Inserir Textos	Ver Plotagem	Texto	Bylayer	Bylayer	Verificar Configurações na TABELA 1.3

Fonte: Elaborado pelo autor.

## APÊNDICE C

### INKSCAPE: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 1 – Elaboração da ilustração da Planta Baixa

**Tempo Estimado = 02h**

AVALIAÇÃO DE SOFTWARE: Inkscape <b>ROTEIRO PARA ELABORAÇÃO DA PLANTA BAIXA</b>		
ORDEM	TAREFA	OBSERVAÇÃO
1	Abrir Arquivo Modulo1ExplnkPB.dxf no Open Office e exportá-lo com extensão ".svg"	salvar na pasta Curso CAD Livre / Inkscape
2	Abrir arquivo no programa Inkscape	
3	Configurar propriedades do Desenho	Ver TABELA 2.1
4	Escalar a imagem do desenho	Escala 1:50
5	Posicionar a imagem na folha de papel	Ver Plotagem
6	Modificar cor dos contornos da imagem para preto	
7	Modificar espessura dos contornos da imagem	Espessura = 0.01
8	Fazer contorno e preenchimento das paredes	Ver TABELA 2.2
9	Fazer contorno e preenchimento das Aduelas	Ver TABELA 2.2
10	Criar padrão de preenchimento dos pilares	Ver TABELA 2.2
11	Fazer contorno e preenchimento dos pilares	Ver TABELA 2.2
12	Fazer contorno e preenchimento das janelas	Ver TABELA 2.2
13	Fazer contorno e preenchimento das portas	Ver TABELA 2.2
14	Criar padrão de preenchimento da Bancada	Ver TABELA 2.2
15	Fazer contorno e preenchimento da Bancada	Ver TABELA 2.2
16	Fazer sombreamento da bancada	Ver Plotagem
17	Fazer contorno e preenchimento das soleiras	Ver TABELA 2.2
18	Fazer contorno e preenchimento da área de piso da Sala de Rádio	Ver TABELA 2.2
19	Fazer juntas do piso da Sala de Rádio (0.60x0.60)cm	Ver TABELA 2.2
20	Fazer contorno e preenchimento dos equipamentos sanitários	Ver TABELA 2.2
21	Fazer contorno e preenchimento da área de piso do Sanitário	Ver TABELA 2.2
22	Fazer juntas do piso do Sanitário (0.60 x 0.60)cm	Ver TABELA 2.2
23	Fazer padrão de preenchimento da área de piso do Atendimento	Ver TABELA 2.2
24	Fazer preenchimento do piso do Atendimento	Ver TABELA 2.2
25	Fazer juntas do piso do Atendimento	Ver TABELA 2.2
26	Fazer contorno e preenchimento das cadeiras	Ver TABELA 2.2
27	Fazer sombreamento das cadeiras	Ver Plotagem
28	Fazer projeção da cobertura	Ver TABELA 2.2
29	Inserir texto "PROJEÇÃO DA COBERTURA"	Fonte = Sans Altura = 10
30	Inserir texto "PLANTA BAIXA"	Fonte = Sans Altura = 20

Fonte: Elaborado pelo autor.

## APÊNDICE C

### INKSCAPE: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 2 – Elaboração da Ilustração da Fachada Frontal

Tempo Estimado = 45min

#### Tarefa 3 – Elaboração da Ilustração da Fachada Lateral

Tempo Estimado = 45min

AVALIAÇÃO DE SOFTWARE: Inkscape <b>ROTEIRO PARA ELABORAÇÃO DAS FACHADAS</b>		
ORDEM	TAREFA	OBSERVAÇÃO
1	Abrir Arquivo Modulo1ExplnkFch.dxf no Open Office e exportá-lo com extensão ".svg"	salvar na pasta Curso CAD Livre / Inkscape
2	Importar o Arquivo para o Inkscape	
3	Escalar a imagem do desenho	Escala 1:50
4	Posicionar a imagem na folha de papel	Ver Plotagem
5	Modificar cor dos contornos da imagem para preto	
6	Modificar espessura dos contornos da imagem	Espessura = 0.01
<b>FACHADA FRONTAL</b>		
7	fazer contorno e preenchimento da Cobertura	Ver TABELA 2.3
8	Fazer contorno e preenchimento das laterais do Pilar	Ver TABELA 2.3
9	Fazer contorno e preenchimento da região central do Pilar	Ver TABELA 2.3
10	Fazer contorno e preenchimento das Aduelas e Perfil Superior	Ver TABELA 2.3
11	Fazer contorno e preenchimento da Base do Piso	Ver TABELA 2.3
12	Fazer traço da Linha de Terra	Ver TABELA 2.3
13	Fazer composição da Esquadria	Ver TABELA 2.3
14	Inserir texto "FACHADA FRONTAL"	Fonte = Sans Altura = 20
<b>FACHADA LATERAL</b>		
15	fazer contorno e preenchimento da Cobertura	Ver TABELA 2.3
16	Fazer contorno e preenchimento da lateral vista do Pilar	Ver TABELA 2.3
17	Fazer contorno e preenchimento das Aduelas e Perfil Superior	Ver TABELA 2.3
18	Fazer contorno e preenchimento da Base do Piso	Ver TABELA 2.3
19	Fazer traço da Linha de Terra	Ver TABELA 2.3
20	Fazer composição da Esquadria	Ver TABELA 2.3
21	Criar padrão de preenchimento da parede do Sanitário	Ver TABELA 2.3
22	Fazer contorno e preenchimento da parede do Sanitário	Ver TABELA 2.3
23	fazer efeito de degradê sobre a textura da parede do Sanitário	Ver TABELA 2.3

Fonte: Elaborado pelo autor.

## APÊNDICE C

### BLENDER: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 1 – Elaboração do Modelo Geométrico Tridimensional

**Tempo Estimado = 3h 30min**

AVALIAÇÃO DE SOFTWARE: Blender					
ROTEIRO PARA ELABORAÇÃO DO MODELO GEOMÉTRICO TRIDIMENSIONAL					
ORDEM	TAREFA	Dim X	Dim Y	Dim Z	OBSERVAÇÃO
1	Abrir o programa e verificar as configurações	-	-	-	Ver PARÂMETRO I
2	Importar desenho Base = Arquivo Modulo1ExpBlnd.dxf	-	-	-	Usar Padrão Autodesk dxf
3	Posicionar desenho Base na camada indicada	-	-	-	Ver PARÂMETRO II
4	Agrupar todos os objetos do desenho Base	-	-	-	A = Seleciona Tudo
5	Inserir Malha e Modelar o piso externo	11,00	6,50	0,15	Fazer modelagem sobre a Base
6	Posicionar malha do piso externo no Layer indicado	-	-	-	Ver PARÂMETRO II
7	Inserir malha e modelar pisos internos	Atendimento = 4,45 Rádio = 1,17 Sanitário = 1,17	Atendimento = 2,45 Rádio = 1,03 Sanitário = 1,03	Atendimento = 0,05 Rádio = 0,05 Sanitário = 0,05	
8	Posicionar malhas dos pisos internos no layer indicado	-	-	-	Ver Base
9	Inserir Malha e Modelar parede	Varia	Varia	2,40	
10	Posicionar Malha da parede no Layer indicado				Ver PARÂMETRO II
11	Inserir Malha e modelar Aduela 1	0,05	0,13	2,20	
12	Fazer cópias da Aduela 1 e posicionar as peças	-	-	-	Ver Base
13	Inserir Aduelas no Layer indicado	-	-	-	Ver PARÂMETRO II
14	Inserir malha e modelar Aduela 2	0,13	0,13	2,20	Pode ser feito com duplicata da Aduela 1, extrudando a face maior desta no Modo Edit, ou modificando as dimensões da peça no Modo Object
15	Duplicar a Aduela 2 e posicionar as peças	-	-	-	Ver Base
16	Inserir Aduela 2 no Layer indicado	-	-	-	Ver PARÂMETRO II
17	Inserir malha e modelar o Montante Superior Lateral	4,32	0,13	0,15	
18	Duplicar o Montante Superior Lateral e posicionar as peças	-	-	-	Ver Base
19	Inserir malha e modelar o Montante Superior Frontal	0,13	0,88	0,15	Pode ser feito com duplicata do Montante Superior Lateral, extrudando a face menor deste no Modo Edit, ou modificando as dimensões da peça no Modo Object
20	Duplicar o Montante Superior Frontal e posicionar as peças	-	-	-	Ver Base
21	Duplicar Montante Superior Frontal e construir o Montante Superior Interno	0,13	0,74	0,15	
22	Posicionar Montante Superior Interno	-	-	-	Ver Base
23	Inserir Montantes Superiores no Layer Indicado	-	-	-	Ver PARÂMETRO II
24	Inserir malha e modelar o Pilar	0,70	0,70	2,90	1 – usar comando Bevel para chanfrar os cantos; 2 - Ajustar a malha à Base no Modo Edit, tendo como referência o posicionamento dos vértices do polígono 3 – Depois de extrudar o polígono fazer cavidades com Operações Booleanas de Diferença e interseção
25	Duplicar o Pilar e posicionar as peças	-	-	-	Ver Base
26	Inserir Pilares no Layer Indicado	-	-	-	Ver PARÂMETRO II

Fonte: Elaborado pelo autor.

(CONTINUA)

## APÊNDICE C

## BLENDER: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

## Tarefa 1 – Elaboração do Modelo Geométrico Tridimensional (continuação)

(CONTINUAÇÃO)

ORDEM	TAREFA	Dim X	Dim Y	Dim Z	OBSERVAÇÃO
27	Inserir malha (cubo) e modelar a Bancada do Atendimento	2,00	0,55	0,70	1 – Espessura da Bancada = 0,07 2 – Utilizar operação Booleana de Diferença entre 2 prismas para modelar a Peça
28	Inserir Malha e modelar Cobertura	8,60	4,25	0,40	1 – As medidas, "DimX" e "DimY", dadas, referem-se à Vista Superior da Cobertura 2 – A medida "DimZ", dada, refere-se à altura total da cobertura
29	Unir (juntar) peças da Cobertura	-	-	-	
30	Posicionar cobertura	-	-	-	Ver Base
31	Inserir Cobertura no Layer Indicado	-	-	-	Ver PARÂMETRO II
32	Inserir Malha e modelar Caixilho da Esquadria Frontal	0,695	2,20	0,03	1 – As medidas "DimX" e "DimY", dadas, referem-se à Vista Frontal da esquadria 2 – A medida "DimZ", dada, refere-se à Espessura da esquadria 3 - Utilizar Operações Booleanas de diferença e Interseção para fazer o vazado dos caixilhos 4 – Utilizar base do Corte como referência para a Modelagem
33	Inserir Malha e modelar as Venezianas para compor a Esquadria	-	-	-	1 – Utilizar Base do Corte como referência para a Modelagem 2 – Medida para Extrusão da Veneziana = 0,595
34	Unir (juntar) peças que formam o painel de Venezianas	-	-	-	
35	Posicionar conjunto de Venezianas no caixilho	-	-	-	Ver Base
36	Inserir Malha (Plane) e ajustar medidas com o vazado do caixilho superior da E	0,595	1,25	0,00	
37	Selecionar conjunto da esquadria e fazer 3 cópias	-	-	-	
38	Rotacionar as Esquadrias para a posição Vertical	-	-	-	
39	Escalonar uma das cópias da Esquadria, no eixo X, para compatibilizar com medida da Esquadria lateral	1,03	Existente	Existente	Sobrepôr a esquadria a ser escalonada sobre a vista superior da Esquadria lateral para ajustar a medida do escalonamento
40	Escalonar a outra cópia da Esquadria Frontal, no eixo X, para compatibilizar com a medida da Esquadria Interna	0,645	Existente	Existente	Sobrepôr a Esquadria a ser escalonada sobre a vista superior da Esquadria Interna para ajustar a medida do escalonamento
41	Inserir Iluminação ao Modelo (04 Spots e 1 Sol)	-	-	-	1 - Ver posicionamento das lâmpadas no Experimento 2 - Deve ser inserida uma primeira lâmpada e gerada cópias 3 - Ver PARÂMETRO III
42	Inserir lâmpadas no Layer Indicado				Ver PARÂMETRO I
43	Inserir Câmera	-	-	-	Ver posicionamento da câmera no Experimento
44	Inserir Câmera no Layer indicado				Ver PARÂMETRO I

Fonte: Elaborado pelo autor.

## APÊNDICE C

### BLENDER: ROTEIRO PARA ELABORAÇÃO DO EXPERIMENTO

#### Tarefa 2 – Aplicação de Materiais

**Tempo Estimado = 1h 30min**

#### Tarefa 3 – Renderização de Imagens (03)

**Tempo Estimado = 1h**

AVALIAÇÃO DE SOFTWARE: Blender			
ROTEIRO PARA APLICAÇÃO DE MATERIAIS NO MODELO GEOMÉTRICO TRIDIMENSIONAL			
<b>OBSERVAÇÕES GERAIS:</b>			
1 – Assista ao videotutorial de aplicação de Imagem UV (Textura), antes de iniciar os procedimentos			
2 – Capture todas as texturas que serão necessárias aplicar ao modelo no site: < <a href="http://www.cgtextures.com">www.cgtextures.com</a> >, antes de iniciar os procedimentos			
3 – Ao realizar o procedimento de aplicação de textura em um objeto, deixe ativo apenas o Layer no qual o mesmo foi inserido			
ORDEM	LOCAL DE APLICAÇÃO DO MATERIAL	TIPO DE MATERIAL	OBSERVAÇÃO
1	Aplicar a textura do Piso Externo	Bloquete de Concreto	Ver PARÂMETRO IV
2	Aplicar textura do piso da área do Atendimento	Mármore Bege Bahia, ou similar	Ver PARÂMETRO IV
3	Aplicar Textura das Soleiras	Mármore Bege Bahia, ou similar	Ver PARÂMETRO IV
4	Aplicar Textura do piso da Sala de Rádio e do Sanitário	Cerâmica	Ver PARÂMETRO IV
5	Aplicar Textura da Parede	Tijolo Cerâmico Maciço	Ver PARÂMETRO IV
6	Aplicar material das Aduales e dos Montantes Superiores	Alumínio Anodizado Amarelo Canário	Ver PARÂMETRO IV
7	Aplicar Material no Caixilho das Esquadrias	Alumínio Anodizado Azul Real	Ver PARÂMETRO IV
8	Aplicar Material nas Venezianas das Esquadrias	Alumínio Anodizado Natural	Ver PARÂMETRO IV

Fonte: Elaborado pelo autor.

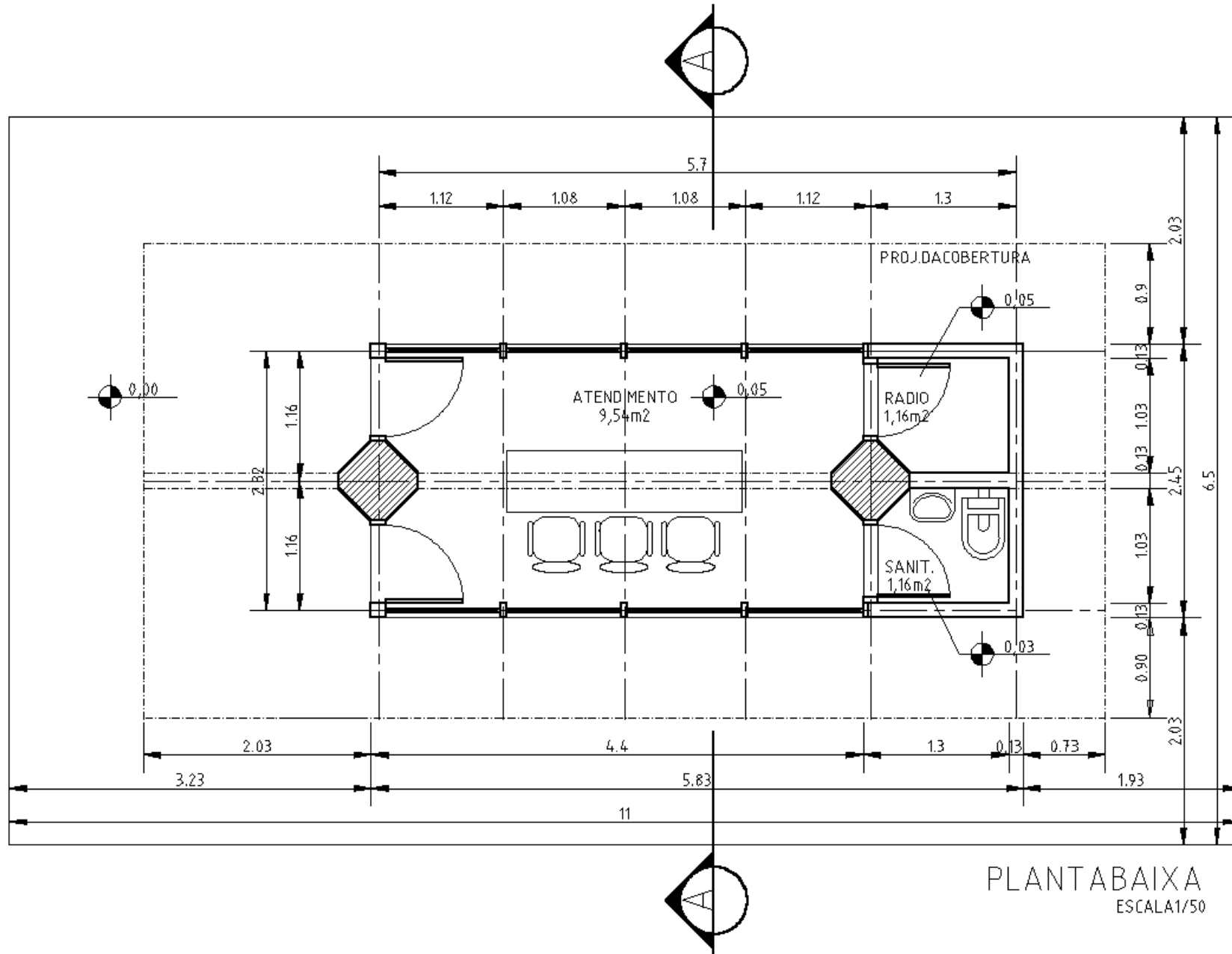
## **APÊNDICE D**

### **RESULTADO EXPERIMENTO PRÁTICO COM O CADUNTUN**



# APÊNDICE D

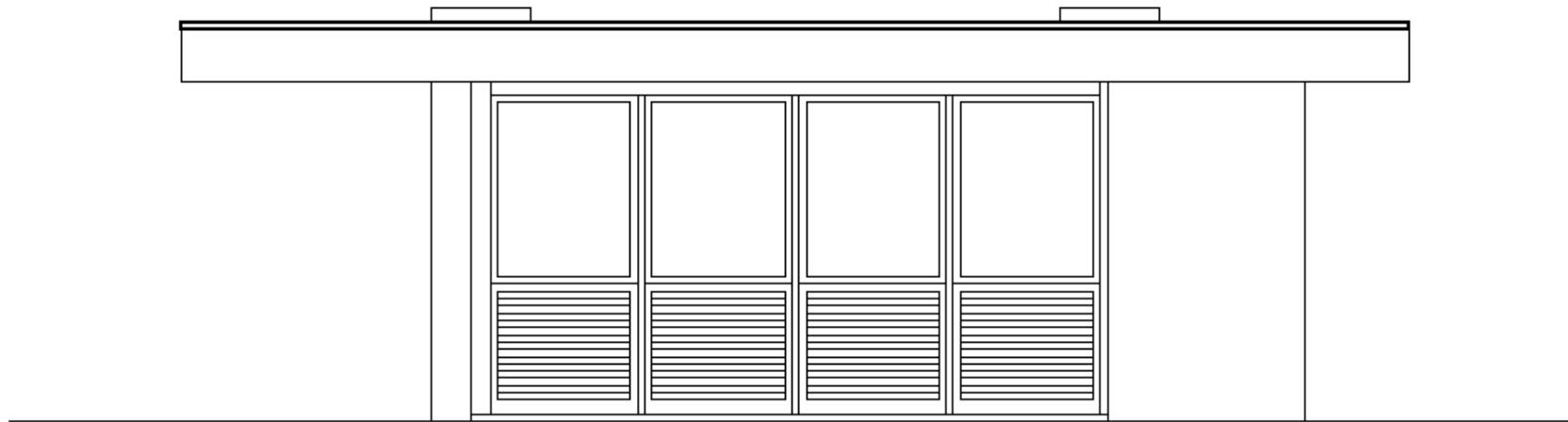
## RESULTADO DO EXPERIMENTO COM O CADUNTUN – PLANTA BAIXA



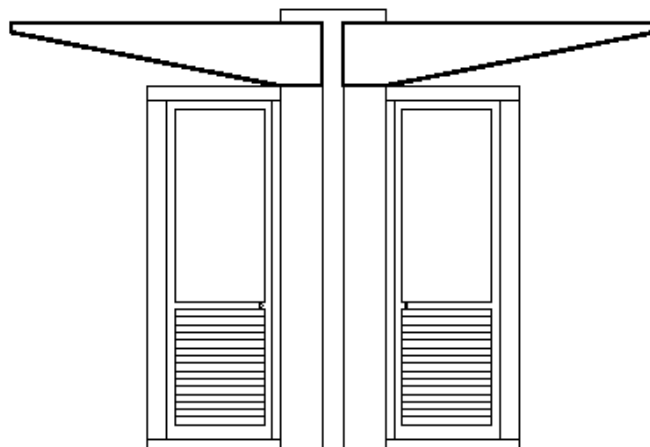
PLANTA BAIXA  
ESCALA 1/50

APÊNDICE D

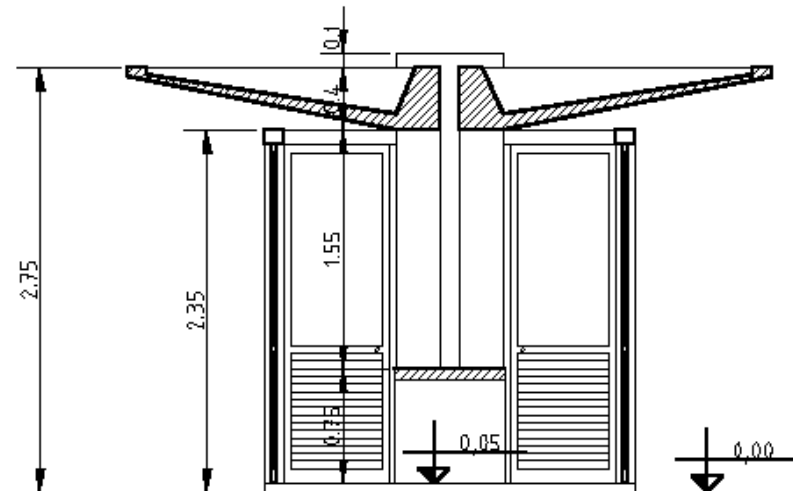
RESULTADO DO EXPERIMENTO COM O CADUNTUN – ELEVAÇÕES



FACHADA LATERAL  
ESCALA 1/50



FACHADA FRONTAL  
ESCALA 1/50

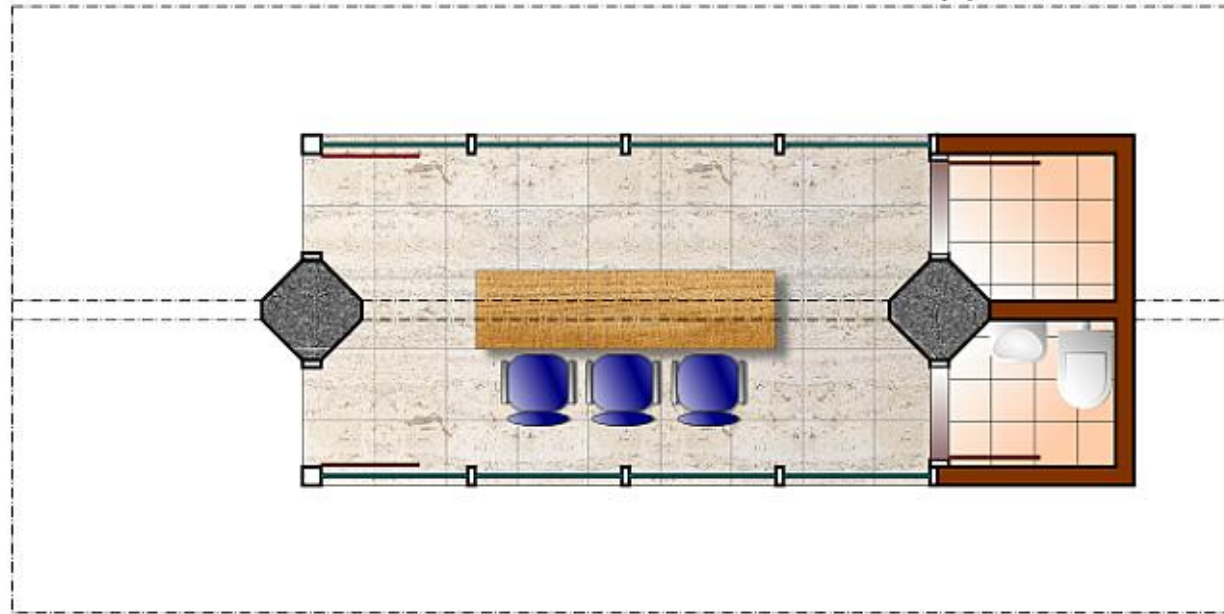


CORTEA-A  
ESCALA 1/50

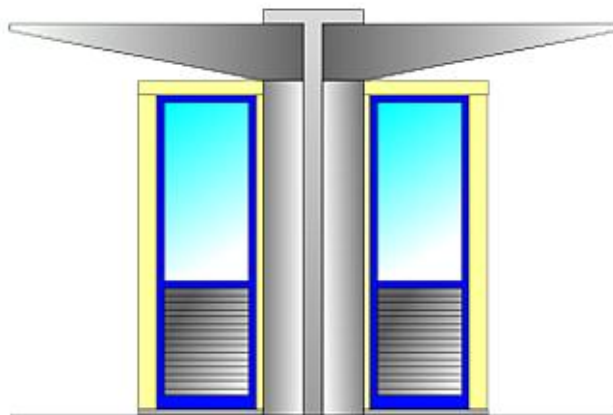
## APÊNDICE E

### RESULTADO DO EXPERIMENTO COM O INKSCAPE

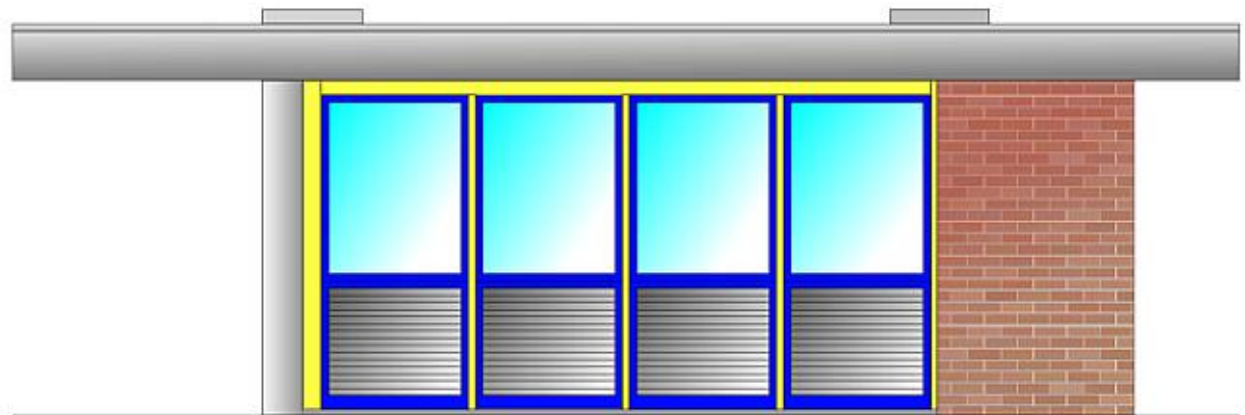
PROJEÇÃO DA COBERTURA



PLANTA BAIXA



FACHADA FRONTAL



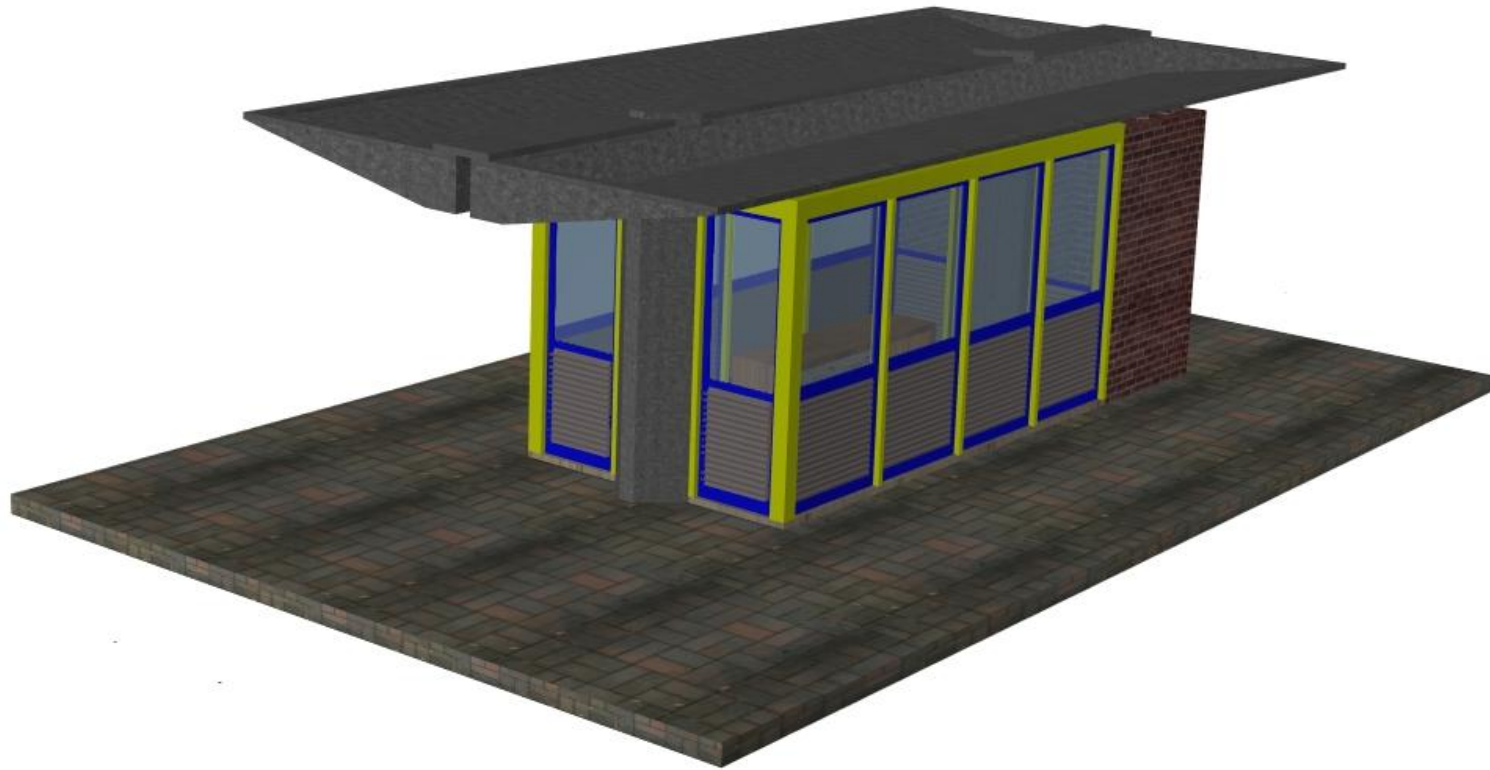
FACHADA LATERAL

## **APÊNDICE F**

### **RESULTADO EXPERIMENTO PRÁTICO COM O BLENDER**

## APÊNDICE F

RESULTADO EXPERIMENTO PRÁTICO COM O BLENDER



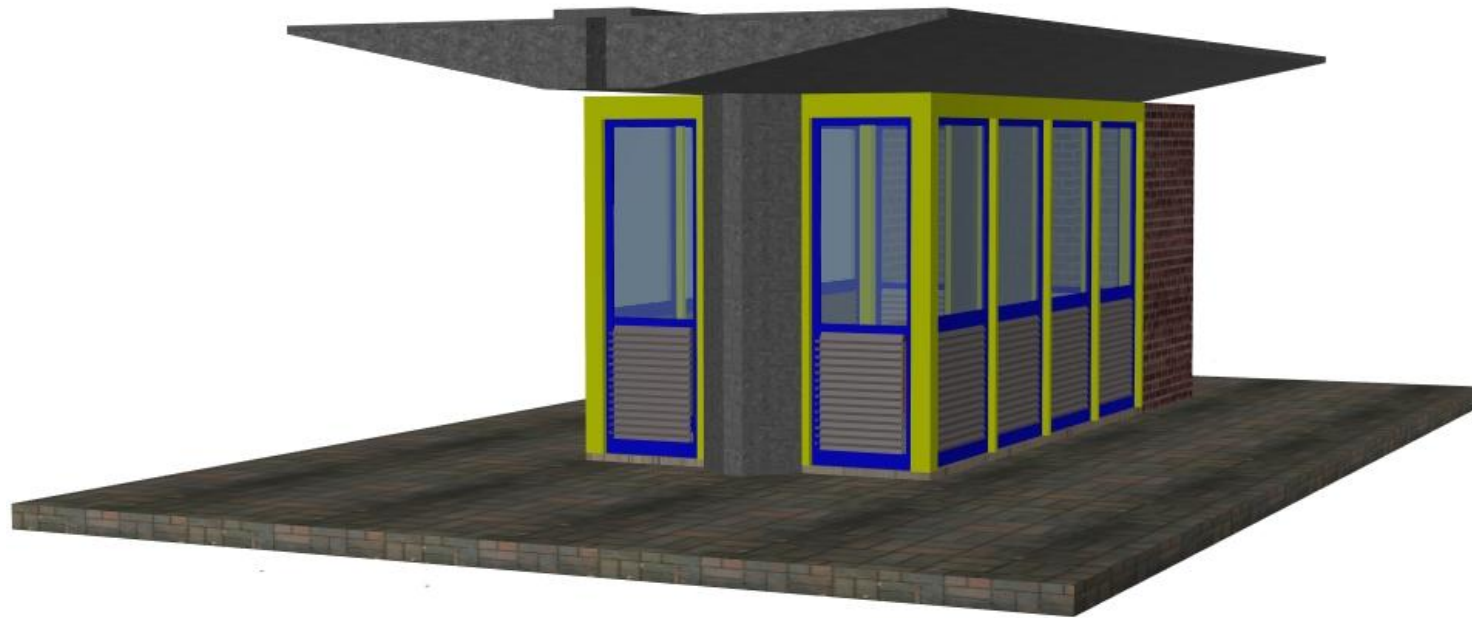
## APÊNDICE F

RESULTADO EXPERIMENTO PRÁTICO COM O BLENDER



## APÊNDICE F

RESULTADO EXPERIMENTO PRÁTICO COM O BLENDER



## APÊNDICE G

### CONFIGURAÇÃO DOS EQUIPAMENTOS UTILIZADOS NO EXPERIMENTO

CONFIGURAÇÃO	LOCAL E PARTICIPANTES DO EXPERIMENTO PRÁTICO									
	Laboratório			Órgão Municipal			Escritório Arquitetura			Residência
	P1	P2	Prof.	P3	P4	Prof.	P5	P6	Prof.	Avaliador
Processador	Intel Pentium Dual 2,00 GHz	Intel Pentium Dual 2,40 GHz	Intel Pentium Dual 2,00 GHz	Intel Pentium Dual 2,20 GHz	Intel Core 2 Duo 2,40 GHz	Intel Pentium 4 1,80 GHz	Intel Pentium 4 2,66 GHz	Intel 1,60 GHz	Intel Pentium 4 2,4 GHz	Intel Core 2 Duo 2,66GHz
Memória RAM	2,00 GB	2,00 GB	2,00 GB	3,00 GB	4,00 GB	1,50 GB	2,00 GB	1,98 GB	2,00 GB	4,00 GB
Sistema Operacional	Windows Vista Business	Windows Vista Business	Windows Vista Business	Windows Vista	Windows 7 Enterprise	Windows XP 2002	Windows 7 Ultimate	Windows Server 2002	Windows 7 Ultimate	Mac OS X 10.5.8
Tipo de sistema	64 Bits	32 Bits	64 Bits	32 Bits	32 Bits	32 Bits	64 Bits	32 Bits	32 Bits	64 Bits

Fonte: Elaborado pelo autor.



**APÊNDICE H**  
**IMODELO DA FICHA DE AVALIAÇÃO**

## APÊNDICE H

### MODELO DA FICHA DE AVALIAÇÃO DO CADUNTUN

Nome do Usuário

---

OBJETIVO/ATRIBUTO	PERGUNTA	MÉTRICA	MEDIÇÕES			
			PLANTA BAIXA	S/N	FACHADAS	S/N
Capacidade de implementar as dimensões especificadas dos elementos arquitetônico	As dimensões especificadas estão implementadas?	Cobertura de implementação das dimensões.	Dist. Entre Eixos		Altura das portas	
			Espess. Da Parede		Desnível do piso	
			Mesa		Nível sup. Do teto	
			Proj. Da Cobertura		Nível sup. Parede	
			Piso Externo		Folha Porta 60	
Capacidade de implementar as tarefas especificadas	As tarefas especificadas foram implementadas com sucesso?	Adequação de implementação funcional	<b>PLANTA BAIXA</b>			
			Nº ORDEM	S/N	Nº ORDEM	S/N
			1		17	
			2		18	
			3		19	
			4		20	
			5		21	
			6		22	
			7		23	
			8		24	
			9		25	
			10		26	
			11		27	
			12		28	
			13		29	
			14		30	
			15		31	
			16			
			<b>FACHADA FRONTAL</b>			
			Nº ORDEM	S/N	Nº ORDEM	S/N
			1		7	
			2		8	
			3		9	
			4		10	
			5		11	
			6		12	
			<b>CORTE A-A</b>			
			13		19	
			14		20	
			15		21	
16		22				
17		23				
18		24				
<b>FACHADA LATERAL</b>						
25		31				
26		32				
27		33				
28		34				
29		35				
30		36				
Capacidade de preservar a integridade do arquivo no caso de ocorrência de falha no sistema	O arquivo manteve a sua integridade após a ocorrência de falha no sistema?	Recuperabilidade de arquivos	Número de falhas no sistema		Número de arquivos perdidos	
Capacidade de evitar a ocorrência de erros	Qual a frequência de erros?	Frequência de erros	Número de falhas no sistema		Tempo	
Capacidade de executar as tarefas especificadas no tempo especificado.	Quanto tempo se demora para executar uma tarefa?	Tempo de tarefa	<b>TAREFA</b>	<b>INICIO</b>	<b>FINAL</b>	<b>TEMPO</b>
			PLANTA BAIXA			
			FACHADA FRONTAL			
			CORTE A-A			
			FACHADA LATERAL			

Fonte: Elaborado pelo autor.

## APÊNDICE H

### IMODELO DA FICHA DE AVALIAÇÃO DO INKSCAPE

Nome do Usuário \_\_\_\_\_

OBJETIVO/ATRIBUTO	PERGUNTA	MÉTRICA	MEDICÕES			
			PLANTA BAIXA	S/N	FACHADAS	S/N
Capacidade de implementar as dimensões especificadas dos elementos arquitetônico	As dimensões especificadas estão implementadas?	Cobertura de implementação das dimensões.	Dist. Entre Eixos		Altura das portas	
			Espess. Da Parede		Desnível do piso	
			Mesa		Nível sup. do Teto	
			Esquadrias Laterais		Nível sup. do Pilar	
			Perímetro Pav Térreo		Nível sup. Da Parede	
Capacidade de implementar as tarefas especificadas	As tarefas especificadas foram implementadas com sucesso?	Adequação de implementação funcional	<b>PLANTA BAIXA</b>			
			Nº ORDEM	S/N	Nº ORDEM	S/N
			1		16	
			2		17	
			3		18	
			4		19	
			5		20	
			6		21	
			7		22	
			8		23	
			9		24	
			10		25	
			11		26	
			12		27	
			13		28	
			14		29	
			15		30	
			<b>FACHADAS</b>			
			Nº ORDEM	S/N	Nº ORDEM	S/N
			1		13	
			2		14	
			3		15	
			4		16	
			5		17	
			6		18	
			7		19	
			8		20	
9		21				
10		22				
11		23				
12		24				
Capacidade de preservar a integridade do arquivo no caso de ocorrência de falha no sistema	O arquivo manteve a sua integridade após a ocorrência de falha no sistema?	Recuperabilidade de arquivos	Número de falhas no sistema		Número de arquivos perdidos	
Capacidade de evitar a ocorrência de erros	Qual a frequência de erros?	Frequência de erros	Número de falhas no sistema		Tempo	
Capacidade de executar as tarefas especificadas no tempo especificado.	Quanto tempo se demora para executar uma tarefa?	Tempo de tarefa	<b>TAREFA</b>	<b>INÍCIO</b>	<b>FINAL</b>	<b>TEMPO</b>
			PLANTA BAIXA			
			FACHADA FRONTAL			
			FACHADA LATERAL			

Fonte: Elaborado pelo autor.

## APÊNDICE H

### MODELO DA FICHA DE AVALIAÇÃO DO BLENDER

Nome do Usuário

---

OBJETIVO/ATRIBUTO	PERGUNTA	MÉTRICA	MEDICÕES			
			PLANTA BAIXA	S/N	FACHADAS	S/N
Capacidade de implementar as dimensões especificadas dos elementos arquitetônico	As dimensões especificadas estão implementadas?	Cobertura de implementação das dimensões.	Cobertura		Altura das portas	
			Espess. Da parede		Desnível do piso	
			Mesa		Nível sup. Do teto	
			Esquadrias laterais		Nível sup. O pilar	
			Esquadrias Frontais		Nível sup parede	
			<b>MODELAGEM</b>			
Capacidade de implementar as tarefas especificadas	As tarefas especificadas foram implementadas com sucesso?	Adequação de implementação funcional	Nº ORDEM	S/N	Nº ORDEM	S/N
			1		23	
			2		24	
			3		25	
			4		26	
			5		27	
			6		28	
			7		29	
			8		30	
			9		31	
			10		32	
			11		33	
			12		34	
			13		35	
			14		36	
			15		37	
			16		38	
			17		39	
			18		40	
			19		41	
			20		42	
			21		43	
22		44				
<b>APLICAÇÃO MATERIAIS</b>						
Nº ORDEM	S/N	Nº ORDEM	S/N			
1		6				
2		7				
3		8				
4		9				
5						
Capacidade de preservar a integridade do arquivo no caso de ocorrência de falha no sistema	O arquivo manteve a sua integridade após a ocorrência de falha no sistema?	Recuperabilidade de arquivos	Número de falhas no sistema		Número de arquivos perdidos	
Capacidade de evitar a ocorrência de erros	Qual a frequência de erros?	Frequência de erros	Número de falhas no sistema		Tempo	
Capacidade de executar as tarefas especificadas no tempo especificado.	Quanto tempo se demora para executar uma tarefa?	Tempo de tarefa	<b>TAREFA</b>	<b>INÍCIO</b>	<b>FINAL</b>	<b>TEMPO</b>
			MODELO			
			GEOM.			
			MATERIAIS			
			RENDER			

Fonte: Elaborado pelo autor.

## **APÊNDICE I**

### **QUESTIONÁRIOS DE VERIFICAÇÃO DO PERFIL DO USUÁRIO**

**APÊNDICE I****QUESTIONÁRIO DE VERIFICAÇÃO DO PERFIL DO USUÁRIO**

**Nome:** (P1)

**Idade:** 20 anos

**Sexo:** M ( ) F ( x )

**1 – Escolaridade/Formação:**

Estudante de arquitetura (5 semestre)

**2 – Atividade Profissional:**

Estudante e estagiária no GEA-Hosp

**3 – Você utiliza softwares gráficos na prática da Arquitetura? Se sim, com qual finalidade você os utiliza?**

Sim, para fazer representação gráfica e maquetes virtuais

**4 – Qual(is) software(s) gráficos você usa com frequência?**

AutoCAD, Revit, Corel Draw, Sketchup

**5 – Qual o estágio de domínio que você possui na operação do(os) programa(s) citado(s) acima: básico; intermediário ou avançado?**

Básico

**6 – Há quanto tempo você usa o (os) programa(s) descrito(s) acima?**

AutoCAD e sketchup: 1 ano e meio

Revit: 1 ano

Corel Draw: 6 meses

**7- Como você aprendeu a utilizar o (os) programa(s) listado(s) acima?**

AutoCad e Revit: Fiz curso

Sketchup e Corel Draw: Aprendi sozinha

**APÊNDICE I****QUESTIONÁRIO DE VERIFICAÇÃO DO PERFIL DO USUÁRIO**

**Nome:** (P2)

**Idade:** 21 anos

**Sexo:** M ( ) F ( X )

**1 – Escolaridade/Formação:**

Ensino Superior (cursando)

**2 – Atividade Profissional:**

Estudante

**3 – Você utiliza softwares gráficos na prática da Arquitetura? Se sim, com qual finalidade você os utiliza?**

Sim. Utilizo softwares gráficos para auxílio na execução de projetos, otimização de tempo e também para visualizar melhor a volumetria dos projetos.

**4 – Qual(is) software(s) gráficos você usa com frequência?**

AutoCAD, sketchup, Photoshop, Corel Draw

**5 – Qual o estágio de domínio que você possui na operação do(os) programa(s) citado(s) acima: básico; intermediário ou avançado?**

AutoCAD: Básico; Sketchup: Básico; Photoshop: Básico; Corel Draw: Básico

**6 – Há quanto tempo você usa o (os) programa(s) descrito(s) acima?**

AutoCAD desde 2009, quando fiz a disciplina informática 1, na faculdade. Sketchup desde 2009. Photoshop desde 2005. Corel Draw desde 2010.

**7- Como você aprendeu a utilizar o (os) programa(s) listado(s) acima?**

Comecei aprender AutoCAD quando cursei a disciplina de informática 1. Sketchup, Photoshop, e Corel Draw, aprendi com auxílio de apostilas da internet.

## APÊNDICE I

### QUESTIONÁRIO DE VERIFICAÇÃO DO PERFIL DO USUÁRIO

**Nome:** (P3)

**Idade:** 29

**Sexo:** M ( ) F ( x )

**1 – Escolaridade/Formação:**

3º GRAU INCOMPLETO/ ARQUITETURA E URBANISMO

**2 – Atividade Profissional:**

Projetos arquitetônicos e urbanísticos.

Plantas decoradas.

Maquetes eletrônicas.

Memorial descritivo.

Cadastro de edificações.

**3 – Você utiliza softwares gráficos na prática da Arquitetura? Se sim , com qual finalidade você os utiliza?**

Sim, para desenvolvimento de projetos arquitetônicos e urbanísticos.

Para apresentação gráfica de projetos.

**4 – Qual(is) software(s) gráficos você usa com frequência?**

SOFTWARE	NIVEL	TEMPO DE USO	COMO APRENDEU
Auto Cad 2011	Avançado	9 anos	Curso e utilizando na faculdade
Corel Draw	Avançado	12 anos	Utilizando no estágio
Photoshop	Intermediário	4 anos	Utilizando no estágio
Sketch Up	Intermediário	2 anos	Curso
3d Studio Max	Básico	1 ano	Apostilas
Revit	Básico	1 ano	Curso
Microsoft Office 2007	Avançado	15 anos	Curso e utilização diária

**OBS:** A respondente sintetizou na tabela acima as respostas referentes às questões 5, 6 e 7.



## APÊNDICE I

### QUESTIONÁRIO DE VERIFICAÇÃO DO PERFIL DO USUÁRIO

**Nome:** (P4)

**Idade:** 32 anos

**Sexo:** M ( ) F ( x )

**1 – Escolaridade/Formação:**

Mestre em História e Teoria da Arquitetura – UPC (Universitat Politècnica de Catalunya)  
Especialista em Planejamento Urbano e Gestão de Cidades – UNIFACS (Universidade Salvador)  
Arquiteta e Urbanista – UNIT (Universidade Tiradentes)

**2 – Atividade Profissional:**

Arquiteta e Urbanista da Prefeitura Municipal de Salvador e autônoma.

**3 – Você utiliza softwares gráficos na prática da Arquitetura? Se sim, com qual finalidade você os utiliza?**

Sim. Os utilizo para o desenvolvimento de projetos arquitetônicos e urbanísticos bem como para a elaboração de projetos de comunicação visual.

**4 – Qual(is) software(s) gráficos você usa com frequência?**

AutoCAD, CorelDRAW, Adobe InDesign, Adobe Photoshop, Autodesk Revit Architecture, 3D Studio Max. Os utilizo muito frequentemente.

**5 – Qual o estágio de domínio que você possui na operação do(os) programa(s) citado(s) acima: básico; intermediário ou avançado?**

AutoCAD – Avançado; CorelDRAW – Intermediário; Adobe InDesign – Intermediário;  
Adobe Photoshop – Básico; Autodesk Revit Architecture – Básico; 3D Studio Max – Básico.

**6 – Há quanto tempo você usa o (os) programa(s) descrito(s) acima?**

AutoCAD – 12 anos; CorelDRAW – 10 anos; Adobe InDesign – 4 anos; Adobe Photoshop – 10 anos; Autodesk Revit Architecture – 1 ano; 3D Studio Max – 12 anos.

**7- Como você aprendeu a utilizar o (os) programa(s) listado(s) acima?**

AutoCAD – curso; CorelDRAW – utilizando; Adobe InDesign – utilizando; Adobe Photoshop – utilizando; Autodesk Revit Architecture – curso; 3D Studio Max – curso.

**APÊNDICE I****QUESTIONÁRIO DE VERIFICAÇÃO DO PERFIL DO USUÁRIO**

**Nome:** (P5)

**Idade:** 20 anos

**Sexo:** M ( ) F ( x )

**1 – Escolaridade/Formação:**

Ensino médio completo; Graduanda no curso de Arquitetura e Urbanismo – UNIFACS.

**2 – Atividade Profissional:**

Estagiária no escritório de arquitetura de Arturo Braga e Carla Macedo Arquitetos.

**3 – Você utiliza softwares gráficos na prática da Arquitetura? Se sim, com qual finalidade você os utiliza?**

Sim, utilizo com a finalidade de realizar trabalhos da faculdade, além do constante uso em estágios realizados na área.

**4 – Qual(is) software(s) gráficos você usa com frequência?**

Utilizo o AutoCAD 2D e SketchUp, com a finalidade do 3D.

**5 – Qual o estágio de domínio que você possui na operação do(os) programa(s) citado(s) acima: básico; intermediário ou avançado?**

No AutoCAD 2D, devido à constante prática, possuo o domínio avançado. O KetchUp, por não ser utilizado com tanta frequência, o domínio é intermediário.

**6 – Há quanto tempo você usa o (os) programa(s) descrito(s) acima?**

Há cerca de 3 anos.

**7- Como você aprendeu a utilizar o (os) programa(s) listado(s) acima?**

Ambos foram lecionados durante o curso de Arquitetura, mas a prática foi adquirida devido à utilização em trabalhos da faculdade e em estágios.

**APÊNDICE I****QUESTIONÁRIO DE VERIFICAÇÃO DO PERFIL DO USUÁRIO**

**Nome:** (P6)

**Idade:** 40

**Sexo:** M ( ) F (x)

**1 – Escolaridade/Formação:**

Nível Superior – Arquiteta Urbanista – Especialista em Arquitetura de Sistemas de Saúde

**2 – Atividade Profissional:**

Sócia Empresa Arturo Braga e Carla Macedo Arquitetos onde desenvolve projetos de arquitetura de diversas ordens

**3 – Você utiliza softwares gráficos na prática da Arquitetura? Se sim , com qual finalidade você os utiliza?**

Sim. Desenvolvimentos de documentos técnicos nas várias etapas de trabalho

**4 – Qual(is) software(s) gráficos você usa com frequência?**

Auto Cad e Corel Draw

**5 – Qual o estágio de domínio que você possui na operação do(os) programa(s) citado(s) acima: básico; intermediário ou avançado?**

Auto-cad avançado e Corel-Draw intermediário

**6 – Há quanto tempo você usa o (os) programa(s) descrito(s) acima?**

15 anos

**7- Como você aprendeu a utilizar o (os) programa(s) listado(s) acima?**

Auto-cad – faculdade e Corel-Draw autodidata.

**APÊNDICE J****MODELO DO QUESTIONÁRIO DE VERIFICAÇÃO DA SATISFAÇÃO DO USUÁRIO****Nome:****1 – Que grau de dificuldade você atribuí ao aprendizado dos comandos do software?**

- a) muita ( )
- b) pouca ( )
- c) nenhuma ( )

**6 – Qual a sua impressão sobre a disposição dos menus na interface do programa?**

- a) ruim ( )
- b) regular ( )
- c) bom ( )
- d) ótima ( )

**4 – Você sentiu dificuldade para aplicação dos comandos e interação com o programa?**

- a) muita ( )
- b) pouca ( )
- c) nenhuma ( )

**2 – Que grau de eficiência você atribui dos recursos oferecidos pelo programa?**

- a) ruim ( )
- b) regular ( )
- c) bom ( )
- d) ótimo ( )

**5 – Como você classifica os resultados obtidos com a aplicação do programa?**

- a) ruim ( )
- b) regular ( )
- c) bom ( )
- d) ótimo ( )

**6 – Você adotaria esse programa para trabalhos futuros?**

- a) sim ( )
- b) talvez ( )
- c) não ( )