



**UNIVERSIDADE DA BEIRA INTERIOR**

DEPARTAMENTO DE INFORMÁTICA

MESTRADO EM ENGENHARIA INFORMÁTICA

**ARMAZENAMENTO E PROCESSAMENTO DE IMAGENS**

**VECTORIAIS EM DUAS DIMENSÕES**

**FILIFE NUNO MATEUS DIAS QUINAZ**

COVILHÃ, JUNHO DE 2008





**UNIVERSIDADE DA BEIRA INTERIOR**

DEPARTAMENTO DE INFORMÁTICA

MESTRADO EM ENGENHARIA INFORMÁTICA

**ARMAZENAMENTO E PROCESSAMENTO DE IMAGENS**

**VECTORIAIS EM DUAS DIMENSÕES**

**FILIFE NUNO MATEUS DIAS QUINAZ**

COVILHÃ, JUNHO DE 2008

Dissertação apresentada à Universidade da Beira Interior para  
obtenção do grau de Mestre em Engenharia Informática

Sob orientação científica de:

Prof. Dr. Pedro Araújo

À minha Mãe e Mafalda

## **AGRADECIMENTOS**

Em primeiro lugar à minha mãe pelo apoio constante.

À Mafalda pelo companheirismo e dedicação.

Ao Professor Pedro, companheiro intelectual e guia académico.

## LISTA DE SIGLAS E ABREVIATURAS

### 2

**2D** - Duas dimensões

### 3

**3D** - Três dimensões

### C

**CAD** - Computer Aided Design

**CGM** - Computer Graphics Metafile

### D

**DSL** - Domain-Specific

**DOM** - Document Object Model

### G

**GPS** - Global Positioning System

### I

**ISO** - International Organization for Standardization

### P

**PDA** - Personal Digital Assistant

**PDF** - Portable Document Format

## **S**

**SVG** - Scalable Vector Graphics

## **U**

**URI** - Uniform Resource Identifier

## Glossário

### E

**Embeber** – Integrar uma parte num sistema circundante

### G

**Glyph** – Componente gráfica que detém a informação sobre a aparência ou forma de um carácter

### H

**Hotspots** – No âmbito deste documento, um Hotspot refere uma área de interacção com o utilizador.

### L

**Layers** – Camadas nas quais são dispostos os elementos gráficos

### R

**Rasterizar** – Converter uma imagem em formato vectorial para um formato raster

**Renderizar** – Processo pelo qual uma imagem é desenhada no ecrã a partir da sua descrição em ficheiro

### S

**Skew** – Transformação na qual se distorce um objecto num determinado ângulo com o objectivo de este parecer inclinado

## V

**Vectorizar** – Converter uma imagem em formato raster numa imagem em formato vectorial

## RESUMO

Este documento representa um estudo acerca do armazenamento e processamento de imagens vectoriais de duas dimensões.

Em relação ao armazenamento, pretendia-se o estudo dos formatos de imagem existentes e a sua estrutura, percebendo quais os mais adequados com vista ao armazenamento a longo prazo. Para tal, efectuou-se uma pesquisa extensa de documentos actuais. Conclui-se deste estudo que todos os formatos possuem algum tipo de vantagem, que justifica a sua existência. Para o armazenamento a longo prazo, verifica-se que este deverá ser efectuado em formatos de codificação em modo de texto e com métodos de compressão não-proprietários. Desta forma foi estudada a estrutura mais detalhada dos formatos SVG, CGM e PDF/A.

No âmbito do processamento de imagem, pretendia-se explorar e testar o processamento do ponto de vista do programador e do utilizador.

Para testar o processamento do ponto de vista do utilizador, foi frequentado um curso pós-laboral de *design* em programas de edição vectorial. O objectivo consistia em verificar em primeira mão as necessidades dos utilizadores. Conclui-se que a simplicidade dos métodos e transparência das funções são factores importantes.

Em relação ao ponto de vista do programador, foi criado um software de edição gráfica de nome DrawIt. O objectivo consistia em testar ferramentas relacionadas com as imagens vectoriais e verificar as dificuldades do processamento do ponto de vista do programador. O software atingiu os objectivos a que se propunha, possuindo no final as capacidades projectadas. Conclui-se que a escolha das ferramentas é um processo importante na construção de um editor gráfico, e que existe uma falta de informação sobre este tipo de sistemas. O DrawIt permitiu o estudo a fundo da norma SVG 1.1, sendo que a sua capacidade de descrever funções através de variados métodos multiplica o esforço do programador. O processamento de imagens vectoriais é exigente, e factores como a criação de threads para procedimentos mais complexos revelam ganhos elevados no tempo de processamento.

Na construção do DrawIt foram estudadas as preferências dos utilizadores face às características dos formatos de imagem, de modo a definir que formato utilizar. Para isso realizou-se um estudo que se mostrou insuficiente. Deste modo, aplicou-se um inquérito de

opinião a profissionais da área. Este inquérito reflecte apenas uma necessidade de um estudo mais aprofundado e abrangente, pois os parâmetros pelos quais muitos programadores se guiam, na escolha do formato vectorial a utilizar, podem, na realidade, estar errados.

Detectou-se ao longo do estudo, uma falta de um *standard* forte na indústria das imagens vectoriais. A ausência deste *standard* levou as empresas a criar muitos formatos similares, dispersando com eles os utilizadores.

Finalmente, foi possível concluir que a escolha do formato de imagem vectorial possui uma enorme relevância no que toca às capacidades descritivas, custo de armazenamento, velocidade de processamento e fidelidade em renderizações futuras.

Existem formatos cuja preservação foi descrita como desejável por outros estudos. Entre os formatos seleccionados encontramos o SVG, o WebCGM e o PDF/A. Estes formatos são também os aconselhados para preservação a longo prazo de imagens.

## Abstract

This document represents a study regarding the processing and archiving of 2D vector images.

Concerning the archiving, the objective was to make a study of the existing vector image formats and their structure, understanding those more adequate for a long term archiving. For that purpose, an extensive research of actual documents was made. The conclusion drawn from this study is that every vector image format existing has some kind of advantage, making its existence justifiable.

The long term archiving should be done using text codification formats associated with non-proprietary compression methods. Taking this into regard, a more detailed study was made about the SVG, CGM and PDF/A formats.

About image processing, the objectives were to explore and test it in two points of view: the programmer's and the user's.

To test image processing in the users point of view, a course of design and vectorial editing programs was taken in a post-employment schedule. The objectives were to verify firsthand the user's necessities. The conclusion is that simplicity of methods and function transparency are major factors.

Regarding the programmers point of view, a graphic editing software named DrawIt was created. The objective was to test tools related to vector images and to verify the difficulties of image processing in the programmers point of view. The software meets the initial objectives. The conclusions are that the tools choice is a very important process in the graphic editing software construction, and that exists a lack of information about these systems. DrawIT allowed a deep understanding of the SVG 1.1 format, being that the SVG's ability to execute the same function using different methods multiplies the programmer's effort. The image processing is demanding, and factors like the creation of threads in more complex procedures result in high gains in processing times.

In DrawIT's construction the user's preferences about image properties were studied, to decide which format to use. To try to achieve that goal, a study of recent documents was made, but revealed to be insufficient. Due to this insufficiency an inquiry of opinion was presented to vector image processing professionals.

This inquiry reflects a necessity of a more profound study, as the parameters that guide many programmer's choice of the vector format, may actually be wrong.

While making the study, a lack of a strong industry standard was detected. The absence of this standard took companies to create many similar formats, causing the dispersion of the final users.

Finally, it was possible to conclude that the choice of the vector image format has a enormous relevance on the description capabilities, storage cost, processing speed and future renderization fidelity.

There are formats which preservation was described as desirable by other studies. Some of the selected formats are SVG, WebCGM and PDF/A. These formats are also adequate for the long term preservation of images.

# ÍNDICE

1. INTRODUÇÃO	1
2. MATERIAL E MÉTODOS	3
3. IMAGEM DIGITAL	6
3.1. Processamento de imagem digital	8
3.2. Imagem vectorial	10
4. ARMAZENAMENTO DE IMAGENS VECTORIAIS	13
4.1. Formatos vectoriais	14
4.2. Que formato escolher?	30
4.2.1. CGM	32
4.2.2. SVG	36
4.2.3. PDF/A	37
4.3. Elementos gráficos vectoriais	39
4.3.1. Primitivas gráficas principais	40
4.3.2. Formas básicas	41
4.3.3. Propriedades dos limites	44
4.3.4. Propriedades das áreas	46
4.3.5. Descrição dos caminhos	48
4.3.6. Atributos de texto	51
4.3.7. Estrutura	53
4.3.8. Transformações e sistemas de coordenadas	53
4.3.9. Clipping	55
4.3.10. Outras propriedades	55
4.4. Formato ASCII e técnicas de compressão	56

5. PROCESSAMENTO DE IMAGENS VECTORIAIS	57
5.1.Preferencias dos utilizadores	59
5.1.1. Fase Metodológica	60
5.1.2. Fase empírica	64
5.2.DrawIt – Descrição e manual do utilizador	70
6. PERSPECTIVAS FUTURAS	85
7. CONCLUSÕES	87

BIBLIOGRAFIA

ANEXOS

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Processos associados à imagem	8
<b>Figura 2.</b> Diferença entre imagens raster e vectoriais	9
<b>Figura 3.</b> Conversões entre os formatos SVG 1.1, WebCGM, PDF e PDF/A	31
<b>Figura 4.</b> Estrutura de um documento CGM	33
<b>Figura 5.</b> Bandeira da Inglaterra em formato vectorial	34
<b>Figura 6.</b> Matriz de transformação de pontos	54
<b>Figura 7.</b> Exemplos de clipping	55
<b>Figura 8.</b> Exemplos de Mascaramento e Filtragem	55
<b>Figura 9.</b> Representação das 6 áreas do interface principal	72
<b>Figura 10.</b> Representação da área 1	72
<b>Figura 11.</b> Menu “File”	73
<b>Figura 12.</b> Interface de abertura de ficheiros	73
<b>Figura 13.</b> Interface de guardar ficheiros no formato SVG	74
<b>Figura 14.</b> Menu “Editar”	74
<b>Figura 15.</b> Menu “Hardware”	75
<b>Figura 16.</b> Interface de controlo de Hardware	75
<b>Figura 17.</b> Menu “Help”	76

<b>Figura 18.</b> Interface de informação do software	76
<b>Figura 19.</b> Área 2	77
<b>Figura 20.</b> Área 3	79
<b>Figura 21.</b> Área 4	79
<b>Figura 22.</b> Área 5	80
<b>Figura 23.</b> Área 6, componente por omissão	81
<b>Figura 24.</b> Área 6, segundo componente	82
<b>Figura 25.</b> Área 6, terceiro componente	83
<b>Figura 26.</b> Visão geral do DrawIT com imagem renderizada	84

# ÍNDICE DE TABELAS

<b>Tabela 1.</b> Primitivas gráficas principais	40
<b>Tabela 2.</b> Descrição das primitivas gráficas principais	41
<b>Tabela 3.</b> Formas básicas	42
<b>Tabela 4.</b> Descrição das formas básicas	43
<b>Tabela 5.</b> Propriedades dos limites	44
<b>Tabela 6.</b> Descrição das propriedades dos limites	45
<b>Tabela 7.</b> Propriedades das áreas	46
<b>Tabela 8.</b> Representação das propriedades das áreas.	47
<b>Tabela 9.</b> Descrição dos caminhos	48
<b>Tabela 10.</b> Representação dos constituintes de caminhos	49
<b>Tabela 11.</b> Atributos de Texto	51
<b>Tabela 12.</b> Descrição dos atributos de texto	52
<b>Tabela 13.</b> Resultados do Inquérito	63

## **1. INTRODUÇÃO**

---

As imagens vectoriais são um subtipo da imagem digital utilizada pelos sistemas informáticos actuais. Com o crescimento dos sistemas informáticos, aparecem novas formas de integrar tecnologias já existentes. Com o desenvolvimento da internet, de sistemas móveis, e sistemas gráficos automatizados surgem necessidades específicas, nas quais a utilização deste tipo de imagens se torna vantajoso.

Neste documento é analisado o estado actual desta tecnologia, juntamente com a sua aplicação no presente e futuro.

O objectivo deste trabalho consiste numa análise aprofundada das características das imagens vectoriais, estudando principalmente formas de armazenamento e o seu processamento.

Devido à grande amplitude de informação relacionada com as imagens vectoriais, o estudo do estado actual da tecnologia consistiu numa análise inicial de carácter geral, especializando-se depois nas propriedades mais relevantes.

Tendo em atenção o armazenamento, o presente documento define imagem vectorial, as suas características, formatos existentes, e comparação dos mais relevantes. Em relação ao processamento, são analisados um estudo de opinião, e a construção de um software com o nome DrawIt.

O estudo de opinião foi realizado em parte devido à falta de informação relacionada com as necessidades dos utilizadores finais desta tecnologia. O conhecimento destas necessidades é importante na selecção do formato a utilizar pelo software a desenvolver. Para concluir o estudo, foi construído um software de edição de imagem baseado no formato SVG 1.1. Este software permitiu estudar de forma aprofundada o formato utilizado, assim como as potencialidades das ferramentas utilizadas.

## **2. Material e métodos utilizados**

---

O estudo tentou ser o mais abrangente possível, tendo sido executado de forma a abranger dois pontos de vista fundamentais: o do programador, e o do utilizador.

Inicialmente houve um estudo teórico das várias propriedades destas imagens. Foram analisados os formatos de imagem mais relevantes da actualidade sendo o estudo aprofundado nos formatos SVG, PDF/A e WebCGM. As formas de armazenamento, foram analisadas tendo em conta vários formatos e os métodos de compressão aconselhados para preservação de longo prazo.

Em relação ao processamento das imagens, realizou-se um estudo bibliográfico sobre as preferências dos utilizadores destes formatos. Devido ao facto de a informação obtida se revelar insuficiente, foi realizado um inquérito de opinião a utilizadores profissionais destas imagens (*vide* ANEXO I). Foram apenas inquiridas pessoas cujo ramo de actividade profissional envolvesse o uso de imagens vectoriais.

Foram também inquiridos alunos de *Design* Multimédia cuja frequência no curso fosse superior ou igual ao 3º ano.

Foi construído um editor gráfico, baseado no formato de imagem vectorial SVG. Este editor tem o nome de DrawIt, permite edição de imagens vectoriais, controlo por voz, rasterização, vectorização e controlo de Hardware.

O software DrawIt foi construído com o ambiente de desenvolvimento Java NetBeans 6.0.1. Foram utilizadas as seguintes ferramentas na construção do software:

- Batik<sup>(37)</sup> : Conjunto de bibliotecas com métodos de criação de DOM's através de código XML para a fácil recolha dos dados. Esta ferramenta suporta também métodos para auxílio na *renderização* de documentos SVG, que embora incompletos, suportam as características principais, incluindo animação.
- Sphinx 4<sup>(38)</sup>- biblioteca para o reconhecimento de comandos de voz.
- Autotrace<sup>(39)</sup>- Software, que a partir de linha de comandos possibilita a vectorização de elementos gráficos *raster* para SVG.

Para melhor compreensão do processamento do ponto de vista do utilizador, foi frequentado um curso de *design* gráfico vectorial, em horário pós-laboral, que decorreu no departamento de Artes e Letras da UBI. Durante o curso de *design* foi explorado principalmente o software Inkscape, cujo formato nativo é o SVG.

### **3. IMAGEM DIGITAL**

---

A palavra imagem provém do latim *Imago*, que significa uma representação visual de um modelo. Este modelo pode ter origem num objecto, numa pessoa, fruto da imaginação do criador da imagem, entre outros.

Associada ao sentido da visão, a imagem é de facto uma forma importante de aquisição de informação, comunicação e até de expressão pessoal ou colectiva.

A forma que a imagem toma varia conforme os suportes de representação (pinturas, fotografias...), os quais têm vindo a evoluir e crescer em número ao longo do tempo.

Com o aparecimento dos sistemas digitais, surgiu também um novo tipo de imagem: a imagem digital. Este tipo de imagem tal como o nome indica, refere-se à representação visual de um modelo através de meios digitais.

Uma imagem digital é constituída por dois tipos de informação, a visual e a descritiva.

A informação visual é o resultado, no caso de um monitor de um computador, da *renderização* da imagem nesse mesmo monitor.

No caso da informação descritiva, esta é referente ao modelo matemático que representa os objectos visualizados.

No seguinte capítulo irão ser abordados os temas de processamento de imagem digital e definição de imagem vectorial.

### 3.1 Processamento de imagem

Quando falamos de processamento de imagem, referimo-nos a operações que são realizadas em imagens que resultam em noutras imagens.

Está associada à alteração da informação visual de uma imagem, que por consequência, necessita da alteração da informação descritiva da mesma.

Associado ao processamento de imagem, temos geralmente a área da visualização, que abrange operações de síntese de imagem, ou seja, de geração visual do modelo.

A figura 1, ilustra os processos fundamentais associados à imagem digital.

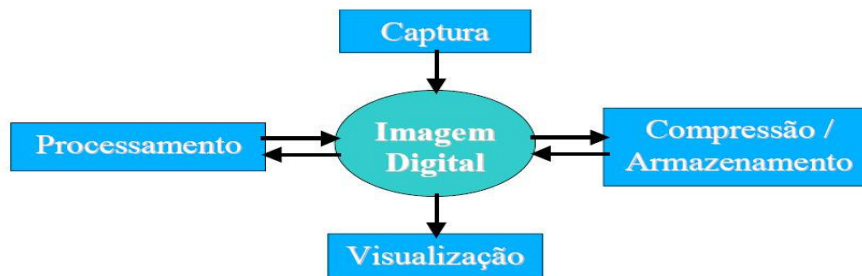


Figura 1: Processos associados à imagem digital<sup>(1)</sup>

A forma como encaramos uma imagem e o seu processamento varia com as necessidades das áreas digitais que utilizam essas mesmas imagens.

Por exemplo, a planta de uma casa pode ser desenhada tendo em conta apenas medidas de distâncias e ângulos, com necessidade de conhecer localizações precisas, podendo ser descrita a preto e branco.

As necessidades alteram-se quando analisamos uma amostra de bactérias num microscópio digital, onde retiramos toda a informação necessária sem conhecer a posição exacta da bactéria na amostra, e onde a existência de cor é fundamental.

A diversidade de objectivos no âmbito da imagem digital levou ao desenvolvimento de modelos de trabalho diferentes.

Assim, a imagem digital divide-se em dois grandes grupos, tendo em conta a forma como estes mesmos grupos descrevem a informação visual da imagem.

São eles: imagens *raster* e imagens vectoriais<sup>(40)</sup>.

De uma forma muito simplista podemos descrever as imagens em formato raster como ficheiros de imagem que utilizam grelhas de pixéis individuais, em que cada pixel pode ter uma cor.

De forma contrária, as imagens em formato vectorial são ficheiros que utilizam relações matemáticas entre pontos para descrever uma imagem. As relações entre formas matemáticas simples como pontos ou linhas, associados entre elas e a formas mais complexas, permitem a descrição de imagens complexas.

Para melhor compreensão da constituição dos dois tipos de formato de imagens segue-se um exemplo descrito na figura 2.

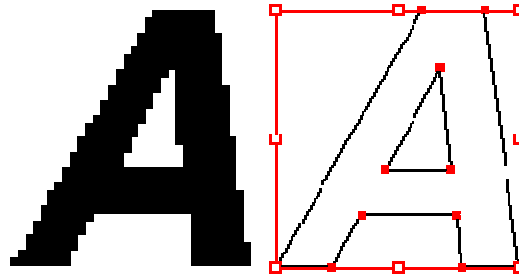


Figura 2 : Diferença entre imagens raster e vectoriais<sup>(40)</sup>

Na figura 2 podemos encontrar do lado esquerdo um exemplo de uma imagem raster constituída por pixéis individuais. Do lado direito vemos uma imagem vectorial, descrita através de formas matemáticas e pontos.

## 3.2 Imagem vectorial

As diferenças entre os dois tipos de formato de imagem manifestam-se principalmente ao nível da definição, também chamado nível de arquivo.

As imagens vectoriais podem reter informação semântica acerca da forma final da imagem e de como ela foi criada. As hierarquias dos pontos, linhas e áreas que definem a imagem são propriedades importantes, que associadas com outras características permitem:

1. Fácil manipulação de formas vectoriais através de fórmulas matemáticas para a obtenção de funções visuais, como por exemplo a rotação
2. *Zoom* não interfere com o detalhe ou qualidade da imagem
3. Objectos constituintes possam ser transformados independentemente
4. Unidades de medida independentes do mecanismo que *renderiza*
5. Texto presente no gráfico possa ser pesquisado como num documento de texto
6. Tamanho geralmente mais pequeno para imagens de grande dimensão
7. Maior velocidade de *renderização*

Outra propriedade importante das imagens vectoriais é que estas podem ser geradas a partir de três processos diferentes:

1. Através da transformação de dados informáticos de forma automática
2. Definindo directamente as curvas e áreas constituintes da imagem
3. Através da extracção de informação de outro tipo de imagens

A segunda forma descrita acima assemelha-se à forma de criação de imagens *raster*. No entanto, o interesse na criação dos formatos vectoriais, surge principalmente devido aos primeiro e terceiro método de geração mencionados.

O primeiro permite uma forma visual de informação, que pode ser gerada automaticamente. Esta forma concretiza-se em sistemas como o desenho dos caminhos gerados pelos GPS's actuais, na criação de gráficos, entre outros.

Em relação à terceira forma, existe um interesse crescente na extracção de informação vectorial de imagens *raster*. Entre os motivos, podemos encontrar a compressão de dados, mas talvez com maior importância, a extracção de informação para pesquisa baseada no conteúdo visual.

A principal diferença no processamento de imagens *raster* e vectoriais, não consta na forma de output final, uma vez que são utilizadas na maioria das vezes, em dispositivos comuns. O que distingue os dois tipos de imagens é a forma como a imagem é criada e modificada.

São considerados os seguintes elementos chave quando se fala em imagens vectoriais<sup>(1)</sup>:

1. Formato de ficheiro
2. Escala (unidade de medida do formato)
3. Precisão e fiabilidade
4. Sistema de coordenadas
5. Geometria (conjunto de elementos geométricos)
6. Relação entre objectos constituintes da imagem
7. Convenções existentes
8. Dados externos associados
9. Metodologia de criação (implicância da recolha de dados para constituição da imagem e consequência na sua precisão e interpretação)

A área de gráficos vectoriais divide-se em 4 grandes áreas.

1. Gráficos 2D. O output de gráficos vectoriais 2D constituídos por pontos, linhas e áreas renderizadas através de um conjunto de propriedades;
2. Gráfico 3D. Semelhante aos gráficos 2D, com a diferença da existência de uma terceira dimensão. Isto resulta na maioria dos casos na necessidade de *renderizar* os gráficos em 2D através de transformações. Para manter a aparência 3D, técnicas como iluminação da cena, adição de sombras e características dos materiais são especificadas;
3. Animação. Adicionar alterações ao longo do tempo na imagem adiciona novas capacidades aos gráficos vectoriais, mas trás também um novo grau de dificuldade;

4. Interacção. A possibilidade de interagir com objectos gráficos adiciona mecanismos de input, um modelo de interacção, gerando também um aumento na complexidade deste tipo de imagem.

Para qualquer uma destas áreas, a existência de poucos *standards*<sup>(1)</sup>, ou afastamento entre eles leva ao reconhecimento da necessidade de maior estudo nesta área.

## **4. ARMAZENAMENTO DE IMAGENS VECTORIAIS**

---

O armazenamento das imagens vectoriais, faz-se através da criação de ficheiros que seguem uma determinada norma. Essa norma, define a codificação tomada (por exemplo modo de texto ou modo binário), informação descrita e extensão de ficheiro.

Assim, é geralmente possível através da terminação do nome do ficheiro, determinar qual a norma utilizada na sua construção. As diferentes normas, servem propósitos distintos, tendo algumas utilizações muito específicas. Neste capítulo, serão abordados alguns dos mais relevantes formatos de imagem vectorial existentes na actualidade, dando especial atenção aos formatos CGM, SVG e PDF/A.

## 4.1 Formatos vectoriais

Existem um número bastante grande de formatos vectoriais, talvez devido em parte à falta de *standards* importantes para este ramo digital.

A grande maioria das empresas que suportam mecanismos de interacção com formatos de imagem vectorial estudadas possuem um formato próprio, e muitas vezes, alteram o formato com novos produtos gerando outro tipo de formato. Exemplo disto é o programa Corel Draw, que por cada versão libertada ao público se fazia acompanhar de uma nova versão do seu formato de imagem.

Um grande número de formatos significa um grande número de formatos pouco importantes, muitos deles até obsoletos.

Segue-se uma lista de formatos de ficheiros de imagens vectoriais, com os formatos mais comuns e relevantes, tendo em conta os *standards*, softwares relevantes do presente e formatos com futuro. Note-se que a seguinte lista servirá apenas como informação básica sobre os formatos apontados, que se expande tendo em conta a importância tanto do formato como das características que o constituem.

### 3DC

Este formato<sup>(2)</sup> representa gráficos vectoriais 3D, mas a grande evolução que este formato ofereceu reside na compressão dos dados *raster* que encapsula.

3DC é um algoritmo de compressão de dados com perda, para mapas normais, inventado e implementado pela ATI.

Está construído com base no algoritmo anterior DXT5 e é hoje em dia um *open standard*. 3DC é agora utilizado tanto pela ATI como pela NVIDIA.

### 3DS

3DS<sup>(3)</sup> é o formato de ficheiros informáticos nativo do software 3D Studio Max que foi substituído mais recentemente pelo 3DS Max cujo formato nativo possui a extensão “.*max*”.

No entanto o 3DS MAX é ainda capaz de ler e escrever ficheiros 3DS.

3DS Max é uma aplicação gráfica que possuiu capacidade de edição 3D desenvolvida pela Autodesk Media and Entertainment. Corre nas plataformas Win32 e Win64, e está desde Abril de 2008 na 11ª versão, que adicionou melhoramentos na velocidade de processamento.

### AI

Adobe Illustrator Artwork (AI)<sup>(4)</sup> é um formato proprietário desenvolvido pela Adobe Systems para a representação de imagens vectoriais de página única, em formato EPS ou PDF. A extensão “.*ai*” é usada então pelo software Adobe Illustrator.

As versões iniciais do formato AI eram verdadeiros ficheiros EPS com restrições de sintaxe, com semântica adicional representada por comentários DSC específicos do Illustrator, que estavam de acordo com a convenção estrutural DSC.

Estes ficheiros eram portanto idênticos aos seus correspondentes Illustrator EPS, mas os *procsets* do EPS (*procedure sets*) eram omitidos do ficheiro e referenciados externamente através de directivas de *include*.

Versões mais recentes do formato AI, incluindo os ficheiros de base PDF e EPS são baseados num formato chamado PGF, que não está relacionado nem com o EPS nem com o PDF.

A compatibilidade com o PDF é atingida não por acrescentar ao formato PDF, mas através de *embeber* uma cópia completa do PGF nativo dentro do ficheiro PDF.

A mesma estratégia é também usada para conseguir gravar a informação nativa do Illustrator em ficheiros compatíveis com EPS.

### **BRL-CAD**

Nome mais comumente dado ao formato de imagem nativo do software com o mesmo nome<sup>(5)</sup>, tem na verdade uma extensão designada “.g”.

O programa BRL-CAD é um sistema CAD, orientado para sistemas 3D, que suporta também outro tipo de formatos vectoriais.

Este sistema foi um dos primeiros sistemas CAD existentes, lançado ao público em 1983 após um início de vida em laboratórios militares.

Foi inicialmente criado para *renderizar* planos de veículos militares, e é a partir de finais de 2004 um projecto *open source*.

### **BSP**

É um ficheiro de descrição de mapas<sup>(6)</sup> usado em jogos como Doom, Quake, e a série de jogos Half-Life .

Estes ficheiros usam partições de espaço binário para gerar níveis, que podem ser *renderizados* rapidamente, enquanto minimizam o número de polígonos que necessitam de ser redesenhados de cada vez que há um refrescamento do ecrã.

Um dos maiores problemas associados com este ficheiro é que o algoritmo de partição de dados não funciona bem com áreas grandes e abertas de jogo.

Ao contrário dos mapas usados no Unreal Tournament, estes ficheiros definem limites dos objectos no mapa, ao invés da definição de áreas ocas.

Estes ficheiros são comumente convertidos de, e para ficheiros “.map”.

Os ficheiros “.bsp”, contêm toda a informação necessária para a definição do mapa à excepção das texturas, que são no caso Quake e Nightfire guardadas em ficheiros como “.pak”, “.pk3”, “.007”, etc.

## CDR

CDR é o formato de imagem nativo do programa Corel Draw, e foi criado pela empresa Corel em 1989.

Este formato é proprietário, o que significa que a sua especificação é divulgada ao público de forma limitada.

O Corel Draw, é um dos melhores softwares de edição de imagem existentes, e compete principalmente com Adobe Illustrator e Macromedia Freehand no que toca a softwares proprietários e com o Inkscape no que toca a freeware.

## CDT

Corel Draw template é um formato nativo do programa Corel Draw. Este formato define templates a serem usadas pelo programa.

## CGM

Computer Graphics Metafile<sup>(7)</sup> foi desenvolvido por especialistas que trabalhavam em comités sob a alçada da International Standards Organization (ISO) e da American Standards National Institute (ANSI). Foi desenhado especialmente para ser um formato independente de plataformas para a transferência de bitmaps e dados vectoriais, que permitisse um variado conjunto de funções de input e output. Apesar de o CGM incorporar extensões desenhadas para suportar imagens *raster*, estes ficheiros são usados principalmente para guardar informação de imagens vectoriais.

Tipicamente contêm dados vectoriais ou *raster*, mas muito raramente contêm os dois ao mesmo tempo.

Permite múltiplas funcionalidades, de modo a tentar suportar as necessidades gráficas de várias áreas profissionais (Artes gráficas, ilustrações técnicas, cartografia, visualização, publicação electrónica, entre outras). Embora este formato possua muitas formas gráficas e atributos, a sua norma é menos complexa do que a norma do PostScript, produzindo ficheiros de menor tamanho, o que torna a transferência e partilha entre sistemas destes ficheiros mais eficaz. No entanto, a sua riqueza de atributos, faz com que uma implementação completa deste formato por parte de programadores de software seja considerada muito difícil. Ainda assim, este formato está numa fase de crescimento no que toca à utilização, sendo uma norma ISO bem estabelecida para a transmissão de dados vectoriais.

## **COLLADA**

Pertence à COLLABorative Design Activity criado para ser um formato de transferência de informação entre aplicações 3D<sup>(8)</sup>.

COLLADA define um *schema* standard de XML e *open source* para a troca de características digitais entre várias aplicações que poderiam gravá-las em formatos incompatíveis entre elas. Estes ficheiros gerados segundo o formato COLLADA são ficheiros XML geralmente com a extensão “.*dae*” (digital asset exchange) .

Originalmente criado pela Sony Computer Entertainment como o formato oficial do desenvolvimento da PlayStation 3 e da PlayStation Portable, pertence agora ao Khronos Group.

## **CSB**

Formato usado pelo Cosmos 3D para guardar gráficos que definem cenas 3D.

O CSB é o formato nativo das aplicações OpenGL Optimizer, e para produzir um ficheiro destes via comandos de programação OpenGL, basta chamar a função `csdStoreFile_csb()`.

## **CSL**

Corel Symbol library é um formato para retenção de informação gráfica de símbolos.

A vantagem de usar símbolos é o aumento não significativo do tamanho do ficheiro, visto que um símbolo é apenas uma referência a um objecto gráfico numa biblioteca.

O Corel Draw tem um sistema de criação de símbolos, que depois são gravados neste formato.

Este tipo de ficheiros pode ser anexado a projectos mais complexos e garante que os símbolos usados na plataforma onde foi criado se mantenham inalteráveis quando transferidos para outras plataformas.

## **DGN**

DGN<sup>(10)</sup> vem da palavra *design*, e é um formato suportado pela Bentley Systems Microstation e pela IGDS (Interactive Graphics Design Systems).

Existem de momento duas versões: a especificação (ISFF) Intergraph Standard File Format conhecida como v7 DGN, publicada nos finais da década de 1980, e a versão actualizada produzida em 2000 também conhecida como V8 DGN.

Ambas as versões são formatos abertos e estão bem documentados pelos seus criadores. São formatos de ficheiro de codificação binária.

## DWF

Design Web Format<sup>(11)</sup> é um formato de imagem desenvolvido pela Autodesk para a distribuição e comunicação eficiente de dados para qualquer pessoa que queira ver ou imprimir ficheiros de desenho. Devido ao facto de estes ficheiros serem altamente compactados, eles são mais pequenos em tamanho, e mais rápidos de transmitir que os ficheiros modificáveis deste mesmo tipo de imagens. Isto deve-se à ausência de cálculos anteriores necessários à abertura dos ficheiros CAD com uma norma complexa.

Com as funcionalidades do DWF, os *designers* podem limitar os dados da imagem para apenas aquilo que querem que o utilizador final veja, podendo também publicar desenhos com várias folhas de vários desenhos AutoCad, num único ficheiro. Podem ainda publicar modelos 3D da maioria das aplicações de desenho da Autodesk.

É importante realçar que os ficheiros DWF não são uma substituição dos ficheiros nativos do AutoCAD (DWG).

O único propósito deste tipo de ficheiros é permitir aos *designers*, engenheiros, projectistas e outros, comunicar informação de *design* e conteúdo para qualquer pessoa que precisa de ver ou imprimir essa informação, sem ter necessidade de ter conhecimentos sobre AutoCad ou outro software de *design*.

O formato é aberto, e a AutoDesk publica a sua especificação assim como disponibiliza bibliotecas em c++ para que os programadores de software possam construir as suas aplicações de modo a suportar o DWF.

O Autodesk Design Review é o software que permite visualizar este tipo de ficheiros e o Autodesk DWF Writer é o programa que permite criar ficheiros DWF a partir de outros ficheiros CAD.

## DWG.

Drawing (DWG)<sup>(12)</sup> é um formato utilizado para descrever imagens em duas e três dimensões nativo originalmente do programa AutoCad, mas que é agora também nativo do Intellicad (e as suas variantes) assim como do PowerCad.

DWG também é suportado de forma não nativa por outras centenas de aplicações de software, o que o torna num formato com grande impacto a nível de número de utilizadores e aplicações.

A AutoDesk que é a empresa que criou o DWG, foi ao longo dos anos criando variantes do formato (18 ao todo), nenhuma das quais com uma especificação disponível ao público.

Trata-se portanto de um formato fechado, que conseguiu no entanto tornar-se num *standard* para intercâmbio de ficheiros CAD entre sistemas.

Estima-se que existam entre 2 a 4 biliões de imagens DWG espalhadas pelo mundo.

A Autodesk proporciona uma biblioteca de leitura e escrita do formato chamada RealDWG através de um processo com critérios rigorosos para empresas que criam aplicações não competitivas.

Existe uma tentativa por parte de várias entidades de realizar *reverse-engineering* do formato.

Para combater este problema há muito existente, a Autodesk criou em 1998, uma função nas suas aplicações que atesta a veracidade dos ficheiros DWG numa resposta a bugs e erros que os utilizadores sofriam por utilizar ficheiros DWG de aplicações concorrentes.

Em resposta, as entidades de *reverse-engineering*, como a mais relevante Open Design Alliance envolveram-se num esforço ainda maior de resolver o problema do DWG e do seu segredo, resultando numa guerra tanto tecnológica como jurídica com a Autodesk.

## DXF

Mais propriamente AutoCAD DXF<sup>(13)</sup> (Drawing Interchange Format, ou Drawing Exchange Format) é um formato de imagem CAD desenvolvido pela Autodesk para permitir a compatibilidade entre o AutoCad e outros programas. A especificação deste formato é disponibilizada à excepção de elementos gráficos mais recentes dos formatos da Autodesk. Estes ficheiros podem ser gravados tanto em formato binário

como em formato ASCII. Este formato visa ser uma reflexão do formato de desenho gráfico do AutoCad, e portanto sujeito a alterações consoante as melhorias no software.

DXF foi originalmente introduzido em Dezembro de 1982, como parte do AutoCAD 1.0, e pretendia ser uma representação dos dados do formato de ficheiros nativo do AutoCAD, o DWG.

Esta necessidade deve-se ao facto de a empresa criadora e que suporta estes formatos não querer publicar as especificações do DWG, revelando ao público apenas a norma relativa ao DXF.

À medida que o AutoCAD se tornou mais poderoso, suportando um maior tipo de objectos, o DXF tornou-se menos útil. Alguns tipos de objectos como alguns tipos de sólidos e regiões, não estão documentados. Outro tipo de objectos estão parcialmente documentados, mas não o suficiente para permitir outros programadores suportá-los.

Quase todas as aplicações comerciais de programadores de software, incluindo a concorrência da Autodesk, escolhem suportar o DWG como o formato principal para trocar dados AutoCad entre sistemas. Isto é conseguido através de bibliotecas da Open Design Alliance, um consórcio industrial não lucrativo, que tem feito *reverse-engineering* no formato DWG.

## EPS

Encapsulated PostScript (EPS)<sup>(14)</sup> é o formato *standard* para importar e exportar ficheiros PostScript. É geralmente um documento PostScript de página única que descreve uma ilustração ou página inteira. O propósito de um ficheiro EPS é ser incluído noutras páginas. Por vezes os ficheiros EPS são designados EPSF, que significa Encapsulated PostScript Format.

Um ficheiro EPS pode conter qualquer combinação de texto, gráficos ou imagens. Uma vez que é na realidade um ficheiro PostScript, é considerado um dos mais versáteis formatos de ficheiro existente. Geralmente contém uma pequena imagem de pré-visualização que é usada para verificar o conteúdo do ficheiro sem o processamento total do mesmo, e sem uma implementação de um interpretador PS. Como é natural, a qualidade sem interpretador PS, ou seja, a qualidade de *renderização* de uma imagem EPS apenas conseguida devido à sua imagem de pré-visualização será sempre inferior quando comparada com a qualidade do processamento do ficheiro na sua totalidade, mas pelo menos é melhor que não ter nada.

Ficheiros EPS podem ser gerados pela maioria das aplicações, o que o torna numa forma segura de transmissão de imagens em diversas plataformas.

No entanto, o EPS está rapidamente a tornar-se desactualizado, uma vez que está a ser substituído pelo PDF, como também o PS em si está a ser substituído também pelo PDF.

Segundo Dov Isaacs da Adobe em relação ao futuro do PostScript “ ...Adobe will continue to support EPS as a legacy graphics format for import of non-color managed, opaque graphical data into Adobe applications (such as InDesign and Illustrator). Although we certainly do not recommend that new graphical content be stored in EPS format (except to satisfy the need to import data into page layout programs that aren’t quite PDF-centric — no need to mention names here!), our user base should feel comfortable that there is no need to worry about a need to convert their very sizable libraries of EPS-based graphic assets.”

## FIG

É um formato de descrição nativo do programa XFig<sup>(15)</sup>.

Xfig é um editor gráfico vectorial *open source* para Windows e maioria das plataformas UNIX.

O XFig é comparado com a concorrência, um editor limitado, suportando apenas as formas vectoriais básicas, assim como imagens *raster*.

Existem vários conversores deste formato para outros mais conhecidos.

A grande vantagem deste editor é ser gratuito para o utilizador, no entanto, esta característica é hoje em dia possuída por softwares superiores como o Inkscape, que torna o XFig, e conseqüentemente o formato “.fig” pouco relevantes no mundo actual.

## GEM VDI

Um formato<sup>(16)</sup> que teve um grande suporte, associado com o GEM GUI da Digital Research. Suporta as formas mais simples, curvas de Bezier e bitmaps *embebidos*.

Apesar de muitas vezes chamado “GEM vector format”, GEM VDI é na realidade um formato *metafile* e está proxicamente associado com o funcionamento da interface de utilizador GEM.

O sistema GEM disponibiliza um driver de *metafile* que é acedido dentro do sistema de programação GEM através de uma API documentada. Pedidos de *renderização* para o driver resultam em itens a serem escritos num buffer no formato do GEM. Elementos do formato consistem portanto em chamadas de funções no sistema de *renderização* do GEM.

O GEM VDI foi originalmente criado na Digital Research, que pertence agora à Novell, sendo que o GEM VDI é suportado agora pela DISCUS Distribution Services.

## Gerber

Um formato<sup>(17)</sup> de ficheiro *standard* usado para placas de circuitos impressos, e programas de *design* industrial. Devido à sua finalidade o formato Gerber possui informação sobre padrões, buracos a serem furados na placa, cortes e polimentos. Este formato foi expandido e chama-se agora também RS-274-X.

Curiosamente existem várias extensões de ficheiro que correspondem ao formato Gerber, e são elas “.gbr”, “.gbx”, “.phd”, “.spl”, “.art”, “.top” e “.bot”.

## HPGL

HPGL<sup>(18)</sup>, algumas vezes referido como HP-GL, é a principal linguagem de controlo de impressoras usada pelas plotters da Hewlett-Packard.

O nome significa Hewlett-Packard Graphics Language e tornou-se um *standard* para quase todas as plotters. As impressoras da HP para além do HPGL suportam também o formato PCL.

A Hewlett-Packard encontra-se na posição privilegiada de ter criado dois *standards* para sistemas de output, HPGL (que significa Hewlett-Packard Graphics Language) e o PCL (que significa Printer Control Language). Apesar do HPGL ter sido criado com o intuito de ser usado para plotters de caneta, variantes do formato encontraram caminho para impressoras inkjet e laserjet.

O PCL, por outro lado, é estritamente usado para dispositivos de input em formato *raster* e está a perder terreno face ao PostScript e Windows-based direct-GDI.

HPGL possui um conjunto de comandos *embebidos* na ROM das plotters de caneta para ajudar a reduzir o trabalho necessário pelas aplicações e seus programadores para criar output para essas plotters. Evoluiu ao longo dos anos à medida que a HP foi criando outros tipos de plotters e impressoras.

No entanto, existe uma desvantagem principal relacionada com este formato e que deriva de ser muito pesado quando comparado com outras linguagens de plotters, o que significa que demora mais tempo a transmitir um HPGL do que por exemplo um DM/PL (digital microprocessor plotting language). Para ultrapassar essa limitação, a HP introduziu o HPGL/2, que continha codificações que permitiam a compressão dos dados reduzindo o seu tamanho consideravelmente. Esta inovação reduziu o tempo de comunicação com a plotter em dois terços.

Com o HPGL a ser o *standard* no que tocava a plotters, os competidores da HP não tardaram a inserir um dialecto semelhante nas suas plotters. Em adição o HPGL tornou-se uma forma de transferência de dados CAD entre diferentes sistemas, uma vez que todos os pacotes CAD produzem HPGL, e alguns lêem o formato.

## HPGL/2

Sucessor do HPGL<sup>(18)</sup>, é menos pesado, mais rápido de processar e de transmitir, poupando, no caso da transmissão dois terços do tempo do seu antecessor.

Este formato destina-se principalmente à interface com o hardware e é muito utilizado visto ser o mais próximo da sua imagem em papel.

É portanto fácil para programadores de interface com o Hardware utilizar este formato, sucessor do *standard* para essas comunicações.

Não é facilmente editado, visto existirem poucas ferramentas de edição disponíveis para o utilizador final, no entanto estão disponíveis múltiplas ferramentas de conversão.

## IGES

Formato<sup>(19)</sup> Vectorial para guardar modelos de malha que representam modelos 2D ou 3D.

Este formato de ficheiros guarda a informação em formato de texto ASCII que facilita a transferência de dados entre diferentes programas e plataformas.

Geralmente os ficheiros IGES utilizam a extensão “.igs” .

Utilizado para intercâmbio de informação digital entre sistemas CAD.

## OBJ

OBJ<sup>(20)</sup> é um ficheiro de definição geométrica desenvolvido inicialmente pela Wavefront Technologies para o seu pacote de animação Advanced Visualizer.

O formato define objectos 3D, nomeadamente a posição de cada vértice, coordenadas das texturas associadas a cada vértice, a normal a cada vértice e as faces que fazem cada polígono.

Esta especificação é aberta e tem sido adoptada por outros programas 3D tais como Poser, Maya, Softimage|XSI, Blender, MeshLab, Misfit Model 3D, 3D Studio Max, Rhinoceros 3D, Hexagon, Newtek Lightwave, Art of Illusion, GLC\_Player etc.

## PDF

O Portable Document Format (PDF)<sup>(21)</sup> foi criado pela Adobe Systems em 1993 para intercâmbio de documentos. Utiliza representação de duas dimensões de forma independente do software, hardware e sistema operativo. Cada documento PDF é uma descrição completa de um documento 2-D (e, com Acrobat 3D, descrição de documentos 3D) que inclui texto, *fonts*, imagens *raster* e gráficos vectoriais.

PDF é um *standard* aberto que recentemente deu um passo grande no sentido de se tornar o ISO 32000.

Quando o PDF saiu para o publico no inicio dos anos 1990s, a sua adopção por parte do publico geral foi lenta. Nessa altura, as ferramentas de criação de documentos PDF (Acrobat) e o sistema de visualização e impressão tinham que ser comprados. As versões iniciais do PDF não suportavam hiperligações, reduzindo a sua utilidade na World Wide Web. O tamanho adicional do PDF comparado com texto simples significava tempos de transferência superiores, o que em modems mais lentos era um problema.

Adicionalmente existia o tempo de processamento de mais informação e formatos concorrentes como o Envoy, Common Ground Digital Paper e até o PostScript da própria Adobe (.ps).

A reviravolta na popularidade do PDF deu-se quando a Adobe começou a distribuição do Acrobat Reader (agora Adobe Reader) sem custo para o utilizador e continuou o suporte ao PDF original que mais tarde se tornou o de facto *standard* para documentos na Web para impressão.

O formato PDF sofreu várias alterações à medida que novas versões do Adobe Acrobat eram produzidas, num total de oito até à data.

## PDF/A

Formato<sup>(22)</sup> de ficheiro electrónico para preservação de gráficos vectoriais de longa duração.

Este formato deriva do PDF e é um standart desde Setembro de 2005.

Pretende representar documentos de tal forma que a sua aparência visual se mantenha independentemente do sistema que realiza o output. O formato PDF não garante a fiabilidade do documento a longo termo. Ou seja, se criarmos uma imagem numa plataforma, nada garante que com o formato PDF/A seja renderizado da exactamente da mesma forma noutra plataforma ou ao longo do tempo.

Este formato exclui capacidades do PDF como som, transparência, vídeo, entre outros. Todas as funcionalidades incluídas num ficheiro PDF/A são obrigatórias e todas as *fonts* devem ser *embebidas*.

## PICT

Formato padrão utilizado para armazenar imagens nos sistemas operativos Macintosh anteriores à versão OS X. Suporta informação *raster* e vectorial. Não tem relevância nos dias de hoje, e não se prevê o seu crescimento no futuro.

## PS

PostScript (PS)<sup>(24)</sup> é uma linguagem de descrição de páginas e linguagem de programação usada principalmente na publicação digital.

Com o PS a ser um de facto *standard* para output de impressão, é natural considerar a mesma linguagem para descrever o output para o ecrã (*renderização*) também.

O rápido aumento da capacidade de processamento da CPU, combinado com os interesses no sistema operativo Windows, levaram a várias tentativas para criar um sistema de visualização de ecrã que usasse o PS como a sua tecnologia principal.

A principal vantagem consiste no fácil modo de impressão visto a existência de controladores nas próprias impressoras que facilitam a vida aos programadores.

## SLD-DWG

Usado pela SolidWorks Corporation, significa SolidWorks Drawing. Este formato é proprietário, e nativo do software SolidWorks.

Este software foi criado em 1995, dois anos após a data da criação da empresa. Tornou-se famoso pela facilidade de desenho 3D que proporciona aos utilizadores.

## STEP

É um formato<sup>(25)</sup> de dados neutro desenhado para a troca de informação entre sistemas CAD. É um *standard* multifacetado desenhado para cobrir a maioria das aplicações de engenharia. Utiliza a linguagem EXPRESS para definir os seus modelos.

Possui a capacidade de conter informação não só sobre a descrição gráfica do objecto mas também do historial pelo qual ele foi construído, desde o seu inicio até à produção final.

## SVF

O Simple Vector Format<sup>(26)</sup>, é um formato de gráficos vectorial que suporta hyperlinks e informação de *layers*. Foi desenvolvido em conjunto pela SoftSource e a NCSA para proporcionar um formato vectorial 2d útil para a Web. A especificação é aberta e está disponível para qualquer pessoa utilizar.

Existe um *plugin* para este formato disponível para o Netscape, que permite *embeber* imagens SVF em HTML.

## SVG

É o *standard* do W3C para a descrição de gráficos vectoriais 2D em XML.

A versão actual é a SVG 1.1<sup>(27)</sup> definida em Janeiro de 2003, sendo que a versão 1.2 está em fase de construção.

É um dos formatos de imagem vectorial aconselhado para a Web que é suportado nativamente pelo firefox e através de *plugins* disponibilizados pela Adobe para outros exploradores de internet.

Actualmente numa fase de expansão, este formato é suportado pela maioria dos softwares da área, sendo até o formato nativo de alguns, como é o caso do inkscape.

Este formato permite animações de objectos ou grupos de objectos constituintes da imagem.

### SWF

Este formato<sup>(28)</sup> é amplamente utilizado na internet dos dias de hoje.

Nativo do programa Flash, pode conter informação vectorial e *raster*. Permite também o *streaming* de vídeo e áudio. É um formato aberto, permitindo que qualquer programa possa em teoria exportar ficheiros SWF.

No entanto existe uma norma criada pela Macromedia, empresa que detém controlo sobre o SWF, que proíbe o uso do SWF para a criação de visualizadores deste formato.

Disponibiliza apenas uma porção dos elementos gráficos e propriedades do SVG e do PDF/A. Existe, apesar de ser um formato vectorial, uma resolução limitada a 1/20 de um pixel, não permitindo um *zoom* muito grande.

### tiny SVG

Versão<sup>(29)</sup> simplificada do SVG utilizada para telemóveis e PDA's entre outros.

### U3D

É um formato aberto<sup>(30)</sup>, que regista a informação em codificação binária. Está ainda sob desenvolvimento pela 3D Industry Forum em colaboração com a Intel e a ECMA International. O objectivo do formato é reutilizar informação CAD 3D.

Permitirá compressão de gráficos 3D, *streaming* através da net, e controlo sobre o detalhe da imagem.

### VML

Vector Markup Language<sup>(31)</sup> é uma linguagem XML usada para produzir gráficos vectoriais. Foi submetido em 1998 pela Microsoft para o W3C numa tentativa de se tornar num *standard* da indústria. Este pedido e outros associados às imagens vectoriais por parte de outras empresas levou a que o W3C cria-se um grupo de trabalho que mais tarde produziu o SVG.

Apesar de largamente ignorado pelos programadores, a Microsoft ainda implementou o VML no internet Explorer 5.0, no Microsoft Office 2000 e nas respectivas versões superiores.

### **WMF, EMF, WMZ, EMZ**

Windows Metafile (WMF) <sup>(33)</sup> é um formato de imagens para sistemas Microsoft Windows, originalmente desenhado nos inícios dos anos 1990s que passou a ser menos comum após o crescimento da Web, devido a formatos como o GIF e JPEG.

É um formato vectorial que permite a inclusão de imagens *raster*. Essencialmente, um ficheiro WMF contém uma lista de chamadas a funções que são processadas pela camada gráfica do Windows, GDI, no sentido de *renderizar* a imagem. Como algumas funções GDI aceitam apontadores para funções para controlo de erros, um ficheiro WMF pode conter código executável.

WMF é um formato 16-bit introduzido no windows 3.0. Uma nova versão 32-bit com comandos adicionais chama-se EMF(enhanced Metafile). Uma versão mais compacta deste formato chama-se Compressed Windows Metafile (WMZ) e a versão compacta da versão 32-bit chama-se Compressed Windows Enhanced Metafile (EMZ).

## 4.2 Que formato escolher?

Quando comparamos a vantagem de armazenamento das imagens em formato vectorial com as do formato *raster*, a diferença de tamanho varia consoante o tamanho da imagem *raster*. Como as imagens vectoriais não alteram o tamanho ocupado em disco dependendo do tamanho da imagem final, estas são geralmente mais pesadas que as *raster* quando o output final é pequeno, e mais pequenas quando o tamanho final da imagem é maior.

As imagens vectoriais tomam vantagem quando a informação a descrever pode ser facilmente associada com formas geométricas. Quando esta codificação não é simples, os formatos *raster* levam vantagens.

O ponto onde em termos de espaço no disco compensa ter imagens *raster* ou vector varia com a imagem e o seu tamanho final no mecanismo de output.

No que toca a imagens vectoriais, um estudo realizado<sup>(1)</sup> conclui que o SVG e o CGM são os únicos formatos de imagem não proprietários mais adequados, por motivos de arquivo. Formatos como o Macromedia Freehand(.af), CorelDraw (.cdr), Adobe Illustrator(.ai), Flas(.swf), entre outros não foram recomendados para preservação e deveriam ser convertidos para um formato *standard*. Isto devido às suas perspectivas futuras, estrutura do formato, integridade da informação e suporte fornecido.

O mesmo estudo considerou também o PDF/A, formato desenvolvido pela Adobe com o intuito de se tornar um *standard* no arquivo de imagens vectoriais, concluindo que a sua eficiência ainda não está comprovada, e portanto gera incerteza quanto ao seu futuro. Estudos para verificar se este formato é viável foram pedidos, e estão em curso.

A resposta à pergunta de que formato escolher pode muito resumidamente ser descrita como: depende! Todos os formatos, inclusive aqueles não mencionados neste documento desempenham a sua função, mais ou menos específica, razão que justifica a sua existência.

Os três formatos apontados como viáveis pelo estudo mencionado anteriormente são bem definidos e bem suportados. Dos três, apenas o PDF/A foi construído com o propósito de ser um formato de arquivo. O SVG 1.1 e o WebCGM confiam em ficheiros externos que não são automaticamente contidos no ficheiro. Não existe a possibilidade de incluir *fonts* no CGM em oposição ao SVG que o permite.

A facilidade de transformação entre estes formatos retira algum do peso da escolha de um deles. Ferramentas de conversão são descritas na seguinte figura:

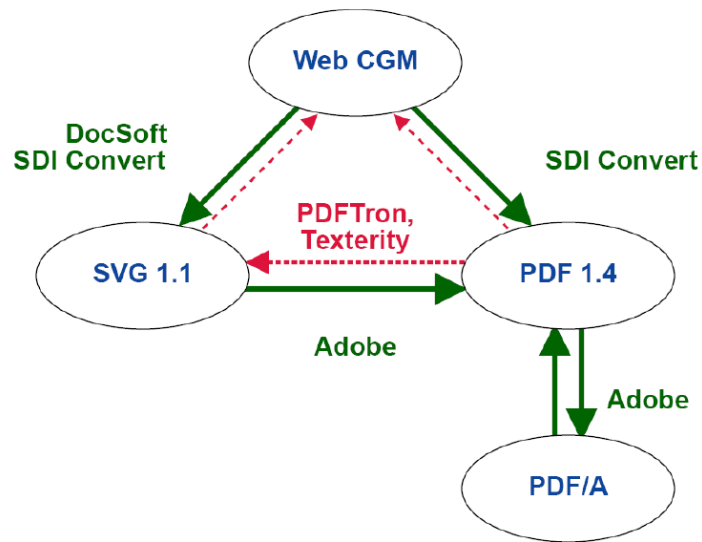


Figura 3 : Conversões entre os formatos SVG 1.1, WebCGM, PDF e PDF/A<sup>(1)</sup>

Para a escolha do formato é necessário ter em conta, em acréscimo a tudo o já referido:

Se a origem dos gráficos é uma área da Engenharia, o CGM é provavelmente a escolha mais robusta. Este formato tem uma facilidade de intercâmbio maior que os outros dois.

O SVG 1.1 é uma aplicação XML que traz consigo um grande conjunto de ferramentas para o auxílio de conversões e transformações.

Se a origem é uma aplicação Adobe, o PDF/A é bem suportado.

## 4.2.1 CGM

Desde o início a interoperabilidade entre os codificadores e decodificadores deste formato tem sido um assunto com a maior importância. Um teste criado pela empresa que suporta o formato, permitiu verificar a fiabilidade das codificações e das decodificações, fazendo com que os problemas fossem reduzidos drasticamente, senão até eliminados no que toca à interoperabilidade<sup>(1,7)</sup>.

Com o crescimento da aplicação deste formato, rapidamente se sentiu a necessidade de criar subtipos do formato de modo a satisfazer as necessidades de várias áreas do mercado.

Isto resultou na *standardização* da noção de perfis de aplicação CGM e no desenvolvimento de perfis específicos para áreas específicas:

1. Perfil ATA: definida para documentação para a manufactura de aviões comerciais.
2. Perfil CALS: (Computer-aided Acquisition and Logistics Support specification) definida pelo ministério da defesa dos EU para ilustrações técnicas e publicações.
3. Perfil CGM\*PIP: para transferência de dados gráficos entre aplicações petro-técnicas na exploração de petróleo e comunidade de produção.
4. Perfil Model: perfil de propósito generalista para aplicações científicas básicas, gráficos técnicos, apresentações, visualização e publicação de documentos.
5. Perfil WebCGM: perfil para uso em aplicações para a WEB.

Os perfis de aplicação descritos acima, definem as opções, elementos e parâmetros necessários para atingir funções particulares.

Tipicamente uma aplicação CGM é baseada num dos perfis descritos acima. Estes perfis, definem diferentes tipos de dados para os parâmetros, dando-lhes também diferentes tamanhos possíveis. Geralmente as aplicações reconhecem valores padrão de um gráfico e adaptam-se de modo a produzir o perfil requerido para satisfazer as necessidades de uma determinada área.

O formato CGM segue uma estrutura ilustrada na seguinte figura:

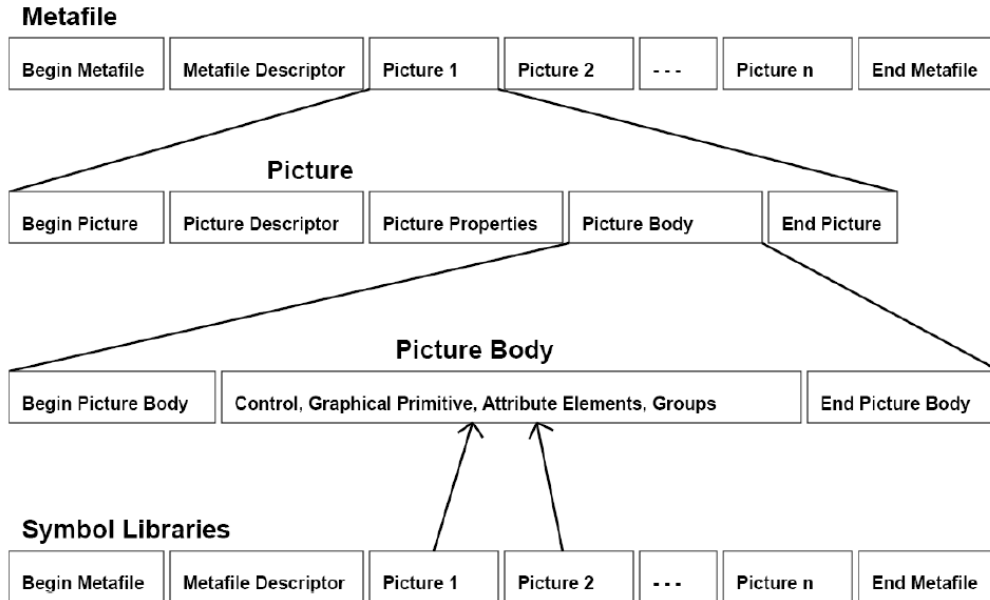


Figura 4: Estrutura de um documento CGM

Tamanho da imagem, escalas e propriedades como grossura das linhas e cor de fundo estão definidas na parte referida na imagem como Picture Descriptor.

Os criadores do CGM oferecem também *pluggins* para acesso à informação visual destes documentos na Web. Como numa fase inicial, os utilizadores de um determinado perfil não podiam ter acesso à informação de outro perfil, surgiu o WebCGM, uma versão mais abrangente de todos os outros perfis.

O WebCGM, provavelmente o mais comum dos perfis contém quatro grupos de elementos gráficos:

- **Grobject**: objecto gráfico com *id* único com possibilidade de link URI. É utilizado para identificar as fontes e destinos dos Hyperlinks.
- **Layer**: Contém uma lista de objectos. Permite que a imagem seja dividida num conjunto de camadas gráficas (*layers*) que podem ser alternadas para ilustrar diferentes partes da ilustração.
- **Para**: define um parágrafo como um conjunto de vários elementos de texto. Os elementos de texto podem estar espalhados pela imagem, mas estão agrupados neste elemento.
- **Sub-para**: um sub-paragrafo usado para identificar fragmentos de texto. Isto serve para especificar por exemplo Hotspots.

Os elementos descritos acima constituem as bases para a pesquisa e realização de ligações entre imagens CGM. Texto internacional utilizado na codificação pode ser Unicode UTF-8 ou UTF-16.

Tal como o formato CGM, o WebCGM é suportado por um pacote de testes extenso, mantido e distribuído pelo CGM Open e WebCGM TC da OASIS.

Segue-se um exemplo de código proveniente de um ficheiro CGM.

```

BEGMF 'UK Flag Clear Text Encoded'; Begin Metafile
MFVERSION 1; Defines the Version of CGM, 1 is the smallest
MFDESC 'RAL GKS 1.11 Cray/COS 88/ 6/17'; Drawn by GKS using a CRAY
VDCTYPE REAL; Real values will be used to define the picture
INTEGERPREC -32768, 32767 Range for integers
REALPREC -8191.0, 8191.0, 8 A default exponent of 8
COLRPREC 255; Colour values in the range 0 to 255
MFELEMLIST 'DRAWINGPLUS'; Defines the set of elements that may appear in the Metafile
BEGPIC 'United Kingdom Flag'; Begin Picture
VDCEXT (0.0,0.2) (1.0,0.8); Min and Max of Coordinates
COLRMODE DIRECT; Colour Specified Directly
BEGPICBODY; Start of Drawing
CLIPRECT (0.0,0.2) (1.0,0.8); Anything drawn outside this area will not appear
INTSTYLE SOLID; Area is filled with a solid colour
FILLCOLR 255 255 255; Fill White
POLYGON (0.0,0.2) (0.0,0.8) (1.0,0.8) (1.0,0.2) (0.0,0.2); Fill whole area white
FILLCOLR 255 0 0; Now fill in Red
POLYGON (0.45,0.2) (0.45,0.8) (0.55,0.8) (0.55,0.2) (0.45,0.2);
POLYGON (0.0,0.45) (1.0,0.45) (1.0,0.55) (0.0,0.55) (0.0,0.45);
POLYGON (0.0,0.755) (0.0,0.8) (0.375,0.575) (0.3,0.575) (0.0,0.755);
POLYGON (0.625,0.425) (0.7,0.425) (1.0,0.245) (1.0,0.2) (0.625,0.425);
POLYGON (0.5750,0.59) (0.575,0.575) (0.625,0.575) (1.0,0.8) (0.925,0.8) (0.575,0.59);
POLYGON (0.0,0.2) (0.075,0.2) (0.425,0.41) (0.425,0.425) (0.375,0.425) (0.0,0.2);
FILLCOLR 0 0 255; Blue
POLYGON (0.1,0.8) (0.425,0.8) (0.425,0.605) (0.1,0.8);
POLYGON (0.0,0.74) (0.0,0.575) (0.275,0.575) (0.0,0.74);
POLYGON (0.575,0.8) (0.9,0.8) (0.575,0.605) (0.575,0.8);
POLYGON (0.725,0.575) (1.0,0.74) (1.0,0.575) (0.725,0.575);
POLYGON (0.0,0.425) (0.275,0.425) (0.0,0.26) (0.0,0.425);
POLYGON (0.1,0.2) (0.425,0.2) (0.425,0.395) (0.1,0.2);
POLYGON (0.725,0.425) (1.0,0.425) (1.0,0.26) (0.725,0.425);
POLYGON (0.575,0.2) (0.575,0.395) (0.9,0.2) (0.575,0.2);
ENDPIC;
ENDMF;

```

O código descrito acima diz respeito à seguinte figura.

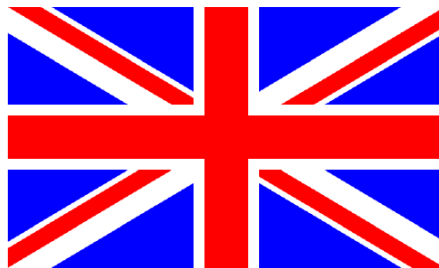


Figura 5: Bandeira da Inglaterra em formato vectorial

Como já foi referido anteriormente o código acima pode não ser válido tendo em conta diferentes perfis do CGM.

Provavelmente o visualizador mais comumente usado para este formato é o plug-in da Micrografx chamado free ActiveCGM . Existe um bom suporte para o WebCGM e é amplamente utilizado em programas de CAD e indústrias aeroespaciais.

## 4.2.2 SVG

O SVG<sup>(1,27)</sup> é um formato mais abrangente que o CGM, com um grande suporte e ainda em crescimento por parte da indústria informática. Possui uma norma extensa que favorece o tamanho das imagens, mas por sua vez dificulta a sua implementação por parte dos programadores.

A versão da bandeira do reino unido mostrada no exemplo do CGM (figura 5) traduzida para SVG seria algo deste tipo:

```
<svg width="600" height="400" xmlns="http://www.w3.org/2000/svg" >
<style type="text/css">
rect {stroke:none;fill:white;}
.red {fill:red;stroke:none}
.blue {fill:blue;stroke:none}
</style>
<g transform="scale(200)" >
<rect x="0" y="0.2" width="1.0" height="0.6" />
<g class="red">
<polygon points="0.45,0.2 0.45,0.8 0.55,0.8 0.55,0.2 0.45,0.2" />
<polygon points="0.0,0.45 1.0,0.45 1.0,0.55 0.0,0.55 0.0,0.45" />
<polygon points="0.0,0.755 0.0,0.8 0.375,0.575 0.3,0.575 0.0,0.755" />
<polygon points="0.625,0.425 0.7,0.425 1.0,0.245 1.0,0.2 0.625,0.425" />
<polygon points="0.5750,0.59 0.575,0.575 0.625,0.575 1.0,0.8 0.925,0.8 0.5750,0.59" />
<polygon points="0.0,0.2 0.075,0.2 0.425,0.41 0.425,0.425 0.375,0.425 0.0,0.2" />
</g>
<g class="blue">
<polygon points="0.1,0.8 0.425,0.8 0.425,0.605 0.1,0.8" />
<polygon points="0.0,0.74 0.0,0.575 0.275,0.575 0.0,0.74" />
<polygon points="0.575,0.8 0.9,0.8 0.575,0.605 0.575,0.8" />
<polygon points="0.725,0.575 1.0,0.74 1.0,0.575 0.725,0.575" />
<polygon points="0.0,0.425 0.275,0.425 0.0,0.26 0.0,0.425" />
<polygon points="0.1,0.2 0.425,0.2 0.425,0.395 0.1,0.2" />
<polygon points="0.725,0.425 1.0,0.425 1.0,0.26 0.725,0.425" />
<polygon points="0.575,0.2 0.575,0.395 0.9,0.2 0.575,0.2" />
</g>
</g>
</svg>
```

No entanto devido às capacidades de caminhos do SVG, processadas através do elemento 'path', podemos descrever a mesma imagem da seguinte forma:

```
<svg width="200" height="120" xmlns="http://www.w3.org/2000/svg" >
<path fill="white" d="M0,0h200v120h-200z"/>
<path fill="red" d="M90,0v120h20v-120zm-90,50h90v20h-90zm110,0h90v20h-90z
m-110,61v9175-45h-15z m125-66h15160-36v-9zm-10,33v-3h10175,45h-15zm-115-78h15170,42v3h-
10z"/>
<path fill="blue" d="M20,120h65v-39zm-20-12v-33h55zm115,12h65l-65-39zm30-45l55,33v-33z
m-145-30h55l-55-33zm20-45h65v39zm125,45h55v-33zm-30-45v39l65-39z"/>
</svg>
```

Apesar de o SVG, tal como o CGM poder ser guardado num documento XML, é normal que grandes ficheiros sejam submetidos a compressão. Este formato aplica o método não-proprietário chamado gzip nos seus documentos. Ficheiros SVG comprimidos têm um ganho de 50% a 80% em relação aos não comprimidos.

## 4.2.3 PDF/A

A informação necessária para reproduzir um ficheiro PDF/A<sup>(1,22)</sup> todas as vezes tem que ser *embebida*, incluindo todas as formas visíveis, como texto, imagens *raster*, objectos vectoriais, *fonts*, informação de cor, entre outras.

O PDF/A, tal como o SVG e o CGM, utilizam primitivas gráficas semelhantes. As técnicas de compressão da informação nos ficheiros são técnicas não proprietárias.

O ficheiro não comprimido da imagem anterior referida na parte dedicada ao CGM representa-se em PDF/A da seguinte forma:

```

12 0 obj
{
0.45 0.2 m
0.45 0.8 l
0.55 0.8 l
0.55 0.2 l
b
0.0 0.45 m
1.0 0.45 l
1.0 0.55 l
0.0 0.55 l
b
0.0 0.755 m
0.0 0.8 l
0.375 0.575 l
0.3 0.575 l
b
0.625 0.425 m
0.7 0.425 l
1.0 0.245 l
1.0 0.2 l
b
0.5750 0.59 m
0.575 0.575 l
0.625 0.575 l
1.0 0.8 l
0.925 0.8 l
b
0.0 0.2 m
0.075 0.2 l
0.425 0.41 l
0.425 0.425 l
0.375 0.425 l
b
}
endobj
13 0 obj
{
0.1 0.8 m
0.425 0.8 l
0.425 0.605 l
b
0.0 0.74 m
0.0 0.575 l
0.275 0.575 l
b
0.575 0.8 m
0.9 0.8 l
0.575 0.605 l
b
0.725 0.575 m
1.0 0.74 l
1.0 0.575 l
b
0.0 0.425 m
0.275 0.425 l
0.0 0.26 l

```

```
b
0.1 0.2 m
0.425 0.2 1
0.425 0.395 1
b
0.725 0.425 m
1.0 0.425 1
1.0 0.26 1
b
0.575 0.2 m
0.575 0.395 1
0.9 0.2 1
b
}
Endobj
```



## 4.3 Elementos gráficos

A grande parte dos formatos vectoriais utiliza um conjunto de propriedades semelhantes para definir os seus gráficos<sup>(1)</sup>.

Devido ao facto de estes serem definidos através de formas geométricas, é natural que muitas dessas formas sejam comuns entre os formatos.

Segundo um estudo realizado, as propriedades dos três formatos de imagem analisados com mais detalhe neste documento representam entre 70% e 80% de todas as características de todos os formatos vectoriais existentes.

Note-se que estas percentagens se tornam ainda mais significativas quando pensamos que existem formatos altamente especializados com propriedades também elas muito especializadas que são utilizadas por um grupo muito diminuto de utilizadores.

As propriedades e elementos gráficos dos formatos são descritos na norma de cada um, sendo cada norma bastante extensa.

Pretende-se neste capítulo descrever as principais primitivas gráficas associadas com os formatos estudados, e a forma como cada um deles as descreve.

Este capítulo diz respeito às normas SVG 1.1, PDF/A e WebCGM 2.0.

### 4.3.1 Primitivas gráficas principais

As primitivas gráficas principais são as formas matemáticas básicas para qualquer documento descrito nos três formatos estudados. São as formas mais poderosas que constituem a base dos documentos mais simples até aos mais complexos.

Descrição	Exemplo
<i>ponto</i>	
<i>caminho aberto</i>	
<i>caminho fechado</i>	
<i>texto</i>	<b>Some text</b>
<i>objecto</i>	
<i>linha de objectos</i>	

Tabela 1: Primitivas gráficas principais<sup>(1)</sup>

- Ponto : Primitiva que identifica um conjunto de coordenadas x,y num gráfico vectorial. Transformações devem permitir que o ponto possa ser identificado após cada transformação.
- Caminho aberto: Primitiva que descreve uma curva podendo conter sub-elementos. O início de Caminho é diferente do ponto final.
- Caminho fechado: Semelhante ao Caminho aberto, mas o ponto final é igual ao ponto inicial.
- Texto: Primitiva que define uma sequência de Glyphs com uma determinada orientação e posição. As características de um Glyph podem ser diferentes dos outros Glyphs do mesmo elemento de texto.
- Objecto: Primitiva que permite *embeber* diferentes elementos gráficos, tais como imagens *raster* e objectos vectoriais. Em relação aos objectos vectoriais, permite que um objecto seja descrito uma vez, e depois instanciado através de vários elementos object variando apenas nas características. Isto permite que o código referente à descrição básica do objecto não seja duplicado. Esta primitiva é principalmente importante quando no caso de arquivo e

transferência de imagens, estas tenham que permanecer inalteradas independentemente do sistema que as interpreta. Isto porque toda a informação necessária à *renderização* tem que ser *embebida* no ficheiro.

- Linha de objecto: Primitiva que define uma sequência de objectos cada um podendo ter características diferentes.

A tabela seguinte refere como se definem as primitivas gráficas principais nos três formatos estudados.

<b>Descrição</b>	<b>WebCGM</b>	<b>PDF/A</b>	<b>SVG 1.1</b>
<b>ponto</b>	polymarker	propriedade de um <b>path</b> primitiva ou objecto	propriedade de um <b>path</b> primitiva ou objecto
<b>caminho aberto</b>	polyline, disjoint polyline circular arc 3 point circular arc centre elliptical arc	path	path
<b>caminho fechado</b>	polygon polygon set	path	path
<b>texto</b>	text restricted text append text	text	text tspan
<b>objecto</b>	polysymbol gobject layer	external	image use
<b>linha de objectos</b>	cell array	inline	symbol svg image (base64)

Tabela 2: Descrição das primitivas gráficas principais<sup>(1)</sup>

Existe uma diferença entre os *standards* na forma em que caminhos mais complexos contendo linhas e curvas são definidos. CGM define linhas e curvas que podem ser definidas como pertencentes a um caminho. O SVG em oposição, define apenas um elemento 'path' que define uma sequência de curvas e linhas que são apenas válidas dentro desse elemento.

A diferença entre o caminho aberto e fechado em SVG é que no último existe um elemento 'z' também designado como fecho de caminho no fim da sequência de formas associadas ao caminho. De forma semelhante o PDF/A define também um elemento de fecho de caminho.

### 4.3.2 Formas básicas

Os três formatos aproveitam-se do facto de muitos gráficos serem simples, consistindo apenas de rectângulos e outros elementos básicos. Na seguinte tabela são evidenciadas as formas básicas utilizadas pelos formatos vectoriais.








<i>Descrição</i>	<i>Exemplo</i>
<i>rectangulo</i>	
<i>circulo</i>	
<i>elipse</i>	
<i>linha</i>	
<i>polyline</i>	
<i>polígono</i>	
<i>arco</i>	

Tabela 3: Formas básicas<sup>(1)</sup>

Apesar destas primitivas poderem ser reproduzidas através das primitivas gráficas definidas anteriormente, a definição de formas básicas favorece as transformações e simplifica o código.

Assim, definem uma série de formas geométricas que reduzem em grande parte o tamanho do código utilizado para as representar, aumentando também a velocidade de processamento.

Na seguinte tabela descrevem-se os elementos utilizados para representar as formas básicas em cada um dos formatos estudados.

<b>Descrição</b>	WebCGM	PDF/A	SVG 1.1
<b>rectângulo</b>	rectangle	re	rect
<b>circulo</b>	circle		circle
<b>elipse</b>	ellipse		ellipse
<b>linha</b>			line
<b>polylinha</b>	polyline		polyline
<b>polígono</b>	polygon polygon set		polygon
<b>arco</b>	arco circular com 3 pontos arco circular centrado arco eliptico		

Tabela 4: Descrição das formas básicas<sup>(1)</sup>

### 4.3.3 Propriedades dos limites

Associadas às formas descritas anteriormente, temos uma série de propriedades associadas aos seus limites, como por exemplo a linha fronteira da forma. Seguem-se as propriedades geralmente disponibilizadas para os limites:

Descrição	Exemplo
largura da linha	
finalização da linha	
junção das linhas	
limite das junções	
padrão de traço	
disposição do traço	
cor	
opacidade	
renderização	

Tabela 5: Propriedades dos limites<sup>(1)</sup>

Uma vez que a propriedade da grossura da linha seja significativa, todas as outras propriedades podem ser aplicadas.

Na seguinte tabela estão descritas as formas de representação das propriedades dos limites pelos três formatos vectoriais estudados.

<b>Descrição</b>	<b>WebCGM</b>	<b>PDF/A</b>	<b>SVG 1.1</b>
<b>largura da linha</b>	line width can be a scale factor edge width	line width	stroke-width
<b>fim da linha</b>	line cap edge cap	line cap	stroke-linecap
<b>junção da linha</b>	line join edge join	line join	stroke-linejoin
<b>limite junção</b>	mitre limit	miter limit	stroke-miterlimit
<b>padrão do traço</b>	line type edge type conjunto de tipos de linha	dash array	stroke-dasharray
<b>disposição do traço</b>	line type continuation line type initial offset edge type continuation edge type initial offset	dash phaset	stroke-dashoffset
<b>cor</b>	line color edge color	strokecolor	stroke
<b>opacidade</b>	RGB ou sRGB-alpha	strokealpha	stroke-opacity
<b>renderização</b>		stroke adjustment flatness	shape-rendering

Tabela 6: Descrição das propriedades dos limites<sup>(1)</sup>

### 4.3.4 Propriedades das áreas

Os principais atributos para o preenchimento de áreas são dados na tabela apresentada em baixo.



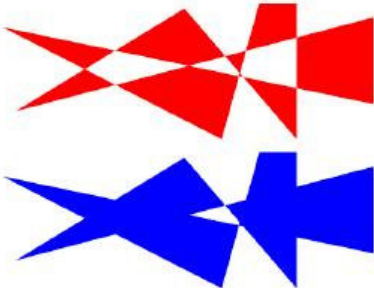
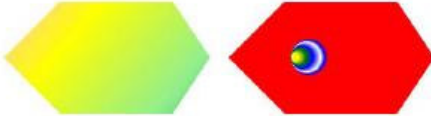
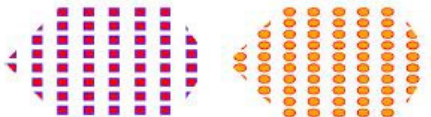
Descrição	Exemplo
cor	
opacidade	
definição de interior	
gradiente	
padrão	
suavidade do gradiente	

Tabela 7: Propriedades das áreas<sup>(1)</sup>

Na seguinte tabela são descritas as formas de representar as propriedades das áreas.

Descrição	WebCGM	PDF/A	SVG 1.1
<b>cor</b>	fill colour	fillcolor	fill
<b>opacidade</b>	RGB ou sRGB-alpha	fillalpha	fill-opacity
<b>definição de interior</b>		evenodd nonzero	evenodd nonzero
<b>gradiente</b>	Definido como parte do sistema de padrões	axial radial Gouraud shading function Coon's patches tensor patches <b>axial</b> between two lines <b>radial</b> between two circles	linearGradient radialGradient
<b>padrão</b>	pattern table pattern size hatch index pattern index interior style tile array	<b>pattern cell</b>	Semelhante ao PDF/A
<b>suavidade do gradiente</b>		smoothness	

Tabela 8: Representação das propriedades das áreas<sup>(1)</sup>

Esta é uma das áreas onde os três formatos mostram menos semelhanças.

Cor e opacidade são semelhantes, mas o preenchimento de gradiente mostra diferenças significativas.

O CGM mostra o menor controlo sobre o processamento de gradientes, enquanto o PDF/A mostra o maior.

### 4.3.5 Descrição de caminhos

Os três formatos permitem especificar uma sequência de comandos de desenho de caminhos. Tanto o PDF/A como o SVG constroem caminhos através do agrupamento de primitivas mais simples. A seguinte tabela mostra os constituintes mais comuns dos caminhos .

Descrição	Exemplo
<b>Comandos de caminhos principais</b>	
<i>move</i> (mover)	
<i>line</i> (linha)	
curvas cúbicas de Bézier	
arco	
fecho de caminho	
<b>Comandos de caminhos secundários para eficiência</b>	
curvas cúbicas de Bézier com o primeiro ponto de controlo igual ao ponto inicial	
curvas cúbicas de Bézier com o segundo ponto de controlo igual ao ponto final	
curvas cúbicas de Bézier com o primeiro ponto a ser o reflexo do ultimo ponto de controlo em relação ao ponto inicial	
curvas quadráticas de Bézier	
curvas quadráticas de Bézier onde o ponto de controlo é o reflexo do ponto de controlo anterior em relação ao ponto inicial	
linha horizontal	
linha vertical	

Tabela 9: Descrição dos caminhos<sup>(1)</sup>

Tanto o SVG como o PDF/A definem os segmentos do caminho tendo em conta um ponto actual, que é actualizado, correspondendo sempre ao último ponto de um caminho.

Na tabela 10 são descritos os constituintes dos caminhos nos três formatos estudados.

Descrição	WebCGM	PDF/A	SVG 1.1
<b>Comandos principais</b>			
<b>mover</b>	ponto inicial de um array	m	M, m
<b>linha</b>	Points after the first in a polyline array	l	L, l
<b>Curvas Cúbicas de Bézier</b>	cubic	c	C, c
<b>arco</b>	circular arc 3 point circular arc centre elliptical arc		A, a
<b>fecho de caminho</b>	connecting edge	h	Z, z
<b>Comandos secundários</b>			
curva cúbica com o primeiro ponto de controlo igual ao ponto inicial		v	
curva cúbica com o segundo ponto de controlo igual ao ponto final		y	
curva cúbica com o primeiro ponto de controlo a ser um reflexo do ultimo ponto de controlo em relação ao ponto inicial	polybezier		S, s
<b>curva de Bézier quadrática</b>			Q, q
curva quadrática com o ponto de controlo a ser reflexo do ultimo ponto de controlo em relação ao ponto inicial			T, t
<b>linha horizontal</b>			H, h
<b>linha vertical</b>			V, v

Tabela 10: Representação dos constituintes de caminhos<sup>(1)</sup>

O comando M (mover) especifica as coordenadas para alterar o ponto actual de um caminho.

O comando L (linha) define uma linha que parte do último ponto do caminho até ao ponto de coordenadas especificadas por este comando.

Os comandos de curvas são constituídos por um ponto inicial (ultimo ponto do caminho), um ponto final especificado pelo comando, e pontos de controlo utilizados para descrever a forma.

### 4.3.6 Atributos de Texto

Os atributos de texto são propriedades que são aplicadas a elementos de texto, descritos na seguinte tabela.

Descrição	Exemplo
font	Times Courier Verdana
tamanho da font	<b>ABC<sub>D</sub></b>
largura da font	ma ma ma
estilo da font	<b>ABC</b>
decoração de texto	<u>underline</u> <u>overline</u> <del>line through</del>
variantes da font	normal SMALL CAPITALS
largura da font	normal <b>bold</b>
alinhamento do texto	left centre right
alinhamento da base do texto	A text middle aligned
disposição da base do texto ( com a tecla shift)	x <sup>super</sup> + y <sub>sub</sub>
modo de escrita	left to right tfel ot thgir down
orientação dos glyphs na vertical	0 06 1 072 d e g 0 0
orientação dos glyphs na horizontal	value:0 < e - c e .. 0 0 value:180 > e - c e .. 0 0
espaçamento dos caracteres	normal and l a r g e r and smaller
espaçamento das palavras	this is normal this is larger
distância entre linhas (altura de cada linha)	this is a typical distance between lines and this is a much larger distance between lines

Tabela 11: Atributos de Texto<sup>(1)</sup>

A maioria dos atributos de texto são semelhantes aos usados em *standards* como XHTML/CSS e Word. A forma de representação dos atributos de texto pode ser encontrada na seguinte tabela.

Descrição	WebCGM	PDF/A	SVG 1.1
<b>font</b>	text font index	Tf	font-family
<b>tamanho da font</b>	character height	Tf	font-size
<b>largura da font</b>	character expansion factor	Th	font-stretch
<b>estilo da font</b>	via font	via font name	font-style
<b>decoreção do texto</b>	text score type	fornecido através de comandos gráficos	text-decoration
<b>variante da font</b>			font-variant
<b>largura da font</b>			font-weight
<b>alinhamento do texto</b>	text alignment		text-anchor
<b>alinhamento da base</b>	text alignment		alignment-baseline
<b>alinhamento da base (pressionando shift)</b>	text alignment	Ts	baseline-shift
<b>modo de escrita</b>	text path	Td	writing-mode
<b>orientação dos glyphs na vertical</b>	character orientation		glyph-orientation-vertical
<b>orientação dos glyphs na horizontal</b>	character orientation		glyph-orientation-horizontal
<b>espaçamento das letras</b>		Tc	character-spacing
<b>espaçamento das palavras</b>		Tw	word-spacing
<b>altura da linha</b>		TI	line-height

Tabela 12: Descrição dos atributos de texto <sup>(1)</sup>

O PDF/A tem a capacidade de processar texto com as características determinadas acima e ainda mais. Este formato não se encaixa sempre bem nesta tabela, pois por exemplo no caso do alinhamento do texto, funciona mais com caixas contentoras do que com o texto em si.

Todos os *standards* oferecem um grande controlo sobre a *renderização* do texto, sendo que os formatos PDF/A e o SVG oferecem um pouco mais que o CGM. Assim, para documentos com elementos gráfico contendo texto e outros, o SVG e o PDF/A serão mais poderosos.

### 4.3.7 Estrutura

Por estrutura do documento<sup>(1)</sup>, entenda-se o mecanismo para construção de objectos mais complexos a partir de outros mais básicos na construção da imagem vectorial total.

O elemento 'g' do SVG permite o agrupamento de primitivas gráficas. Grupos podem conter outros grupos, sendo o desenho uma hierarquia de objectos mais pequenos agrupados em *layers* (grupo que define uma camada).

No SVG a ordem com a qual se agrupam os objectos altera a ordem pela qual eles são *renderizados*. O motivo porque isto acontece é porque os documentos SVG são processados como a leitura de um texto, em que os documentos descritos primeiro são os primeiros a serem *renderizados*.

Transformações aplicadas nos grupos são transportadas para os elementos constituintes através de mecanismos de herança. O mesmo serve para os outros dois formatos através de abordagens ligeiramente diferentes.

### 4.3.8 Transformações e sistemas de coordenadas

Para os formatos definidos as transformações e sistemas de coordenadas são definidas da seguinte forma:

1. Permitem definir os objectos gráficos nas coordenadas de escolha do utilizador
2. Definem um sistema de coordenadas virtual na qual os objectos da cena são transformados
3. CGM: define um sistema de coordenadas virtual por omissão, que é depois alterado consoante o sistema de coordenadas do utilizador
4. PDF/A: define um espaço de utilizador por omissão onde cada valor de coordenada dista do próximo 0,035 cm.
5. SVG: ou define um atributo `viewBox` no elemento raiz do documento que dá a extensão da área gráfica a visualizar pelo utilizador, ou então guia-se pela altura e largura também especificadas no elemento raiz
6. No CGM e no PDF/A o eixo Y é definido de baixo para cima enquanto no SVG é definido de cima para baixo

Estes formatos permitem valores para coordenadas reais, pelo que a precisão não é comprometida apesar de pequenos arredondamentos.

O SVG e o PDF/A partem do princípio que os objectos vão ser *renderizados* no sistema de coordenadas do utilizador, e qualquer transformação é aplicada tendo em conta esse mesmo sistema

As transformações sobre os elementos gráficos são:

1. Translate
2. Scale
3. Rotate
4. Skewx
5. Skewy

Independentemente das transformações e do modo como se processam, estas podem ser descritas através de seis parâmetros na seguinte matriz.

Os parâmetros (a,b,c,d,e) , aplicados no ponto original (xold,yold) transformam-no no ponto (xn,yn). Tal como descrito na figura seguinte, o produto entre a matriz transformação e o ponto antigo (anterior à transformação) tem como resultado uma matriz representativa do ponto novo (após a transformação)

$$\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{old} \\ y_{old} \\ 1 \end{bmatrix}$$

Figura 6: Matriz de transformação de pontos<sup>(1)</sup>

### 4.3.9 Clipping

O Clipping nos gráficos computacionais é usado para restringir o efeito de um ou vários elementos gráficos. Elementos dentro da área de Clipping são desenhados, enquanto que o resto é descartado.

Aplicar áreas de clipping complexas pode resultar em primitivas parciais cuja representação no sistema pode ser difícil. Como consequência é habitual que esta operação seja aplicada nas fases finais do processamento da imagem. Seguem-se alguns exemplos de Clipping:

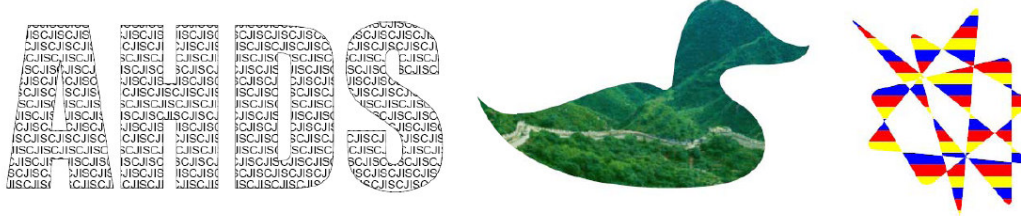


Figura 7: Exemplos de clipping<sup>(1)</sup>

#### 4.3.10 Outras propriedades

A lista de propriedades descrita acima está longe de completa.

Algumas das outras áreas amplamente utilizadas são:

1. Mascaramento
2. Filtragem

Alguns elementos gráficos podem ser usados como uma máscara para outros objectos. Esta máscara altera as características dos objectos mascarados. A filtragem consiste numa série de operações de processamento gráfico que são aplicadas num objecto antes de ele ser *renderizado* no sistema do utilizador. A figura abaixo mostra exemplo destas duas propriedades.



Figura 8: Exemplos de Mascaramento e Filtragem<sup>(1)</sup>

No caso do mascaramento da imagem, o efeito é correspondente à imagem da esquerda. No caso da filtragem da imagem da direita, fora, adicionados ruído, efeito de *blur* e efeitos de luz.

## 4.4 Formato ASCII e técnicas de compressão

Um dos motivos que levou à conclusão que estes formatos deveriam ser preservados, reside no facto de a codificação ser legível independentemente de se ter ou não um interpretador do formato.

O estudo aconselha todos os formatos de imagem vectoriais a codificarem a informação através do sistema ASCII. Apesar das possíveis vantagens, é difícil levar esta tarefa a cabo na prática.

O WebCGM, perfil mais importante do CGM para propósito de arquivo, não suporta a codificação de caracteres do CGM. O CGM é utilizado principalmente para o *debug* e nada mais.

Apesar do SVG poder ser definido em XML sem compressão, não há restrição quanto à codificação, podendo ser ou não ASCII. Da mesma forma que o SVG, o PDF/A não contém nenhuma restrição de codificação.

No entanto, existe uma clara necessidade de compressão por parte destes ficheiros. Assim, uma solução também sugerida pelo estudo consiste na boa documentação de técnicas de compressão para estes formatos usando apenas técnicas de compressão não-proprietárias. Se isso for conseguido, os gráficos vectoriais gravados de forma comprimida poderiam ser aceites da mesma forma que os não comprimidos. Apenas uma fracção muito pequena dos formatos vectoriais existentes são arquivados usando a codificação ASCII.



## **5.PROCESSAMENTO DE IMAGENS VECTORIAIS**

---

Referente ao processamento de imagens, podemos considerar duas áreas fundamentais. O processamento por parte de utilizadores de edição gráfica, e o processamento por parte dos programas que estes utilizam.

Para uma melhor percepção das necessidades dos utilizadores foi frequentado um curso de *design* de imagens vectoriais utilizando principalmente o software Inkscape. Os objectivos consistiam em verificar em primeira mão, as dificuldades e necessidades dos utilizadores. Conclui-se que a simplicidade dos comandos e transparência dos métodos são factores muito importantes na construção do software.

No entanto, esta experiência foi uma experiência pessoal, tendo sido posteriormente realizado um estudo de documentos actuais acerca das necessidades dos utilizadores deste tipo de software.

## 5.1 Necessidades dos utilizadores

A evolução informática tem vindo a conseguir aumentar o poder de processamento, capacidade de armazenamento e taxas de transferência dos ficheiros.

No entanto, em relação às imagens vectoriais, os sistemas informáticos estão ainda longe de conseguir o seu processamento sem um custo relativamente elevado. É precisamente por esse motivo que existem diversos formatos disponíveis, pois uns exibem vantagens, por exemplo, no armazenamento, enquanto outros exibem vantagens no tempo de processamento.

Do ponto de vista do programador, quando desenvolve uma aplicação relacionada com imagens vectoriais, o formato a escolher é essencial. Geralmente os softwares que trabalham com imagens possuem um formato nativo que vai ao encontro às necessidades principais do seu público-alvo.

No caso de um servidor de imagens, é natural que o formato escolhido seja aquele cujo custo de armazenamento seja baixo, com taxas de transferência baixas. Já no caso de editores gráficos, a escolha costuma recair sobre imagens amplamente utilizadas, cuja norma seja completa o suficiente para reduzir tempo de processamento.

Quando se trata de escolher o formato, as características dos formatos existentes são conhecidas, no entanto as necessidades dos futuros utilizadores e as suas preferências são geralmente desconhecidas. O programador escolhe tendo em conta as características do seu programa, tentando equilibrar as vantagens e desvantagens de cada formato. O estudo que se segue seguiu as normas descritas nos livros de investigação científica consultados<sup>(34,35,36)</sup>.

## 5.1.1 FASE METODOLÓGICA

### PROBLEMÁTICA DO ESTUDO

Existe uma notória falta de informação a este nível, pelo que foi tomada a decisão de estudar as necessidades e preferências dos utilizadores de imagem vectorial em Portugal.

Este estudo começou com algumas limitações à partida. O desconhecimento do número de profissionais em cada área das imagens vectoriais, torna difícil o estudo estatístico. No entanto, podemos efectuar um estudo de opinião sobre as preferências de alguns profissionais, não tendo no entanto em conta a sua área de especialização.

De todos os documentos analisados inicialmente, o custo de armazenamento de um determinado formato é muitas vezes mencionado como importante, não tendo no entanto comparações com outras características.

### QUESTÃO DE INVESTIGAÇÃO

O estudo de documentos levado a cabo inicialmente não conseguiu dar uma resposta satisfatória à seguinte questão:

Qual a prioridade a dar a cada uma das características dos formatos de imagem vectorial de modo a escolher o formato mais vantajoso para o utilizador na construção de um software de processamento de imagem?

## OBJECTIVOS

O objectivo consistiu em realizar um estudo de opinião de alguns profissionais da área do processamento de imagem. Este estudo tem como objectivo responder à questão de investigação, sendo esta resposta válida para o grupo avaliado neste mesmo estudo.

### Objectivo geral:

Avaliar, junto de um conjunto de profissionais da área do processamento de imagem, quais as preferências dos utilizadores em relação às propriedades dos formatos.

### Objectivos específicos:

1. Verificar qual a importância dada pelo grupo inquirido a cada uma das propriedades dos formatos de imagem.
2. Tendo em conta as respostas, criar uma lista de propriedades ordenada pela sua ordem de relevância para os utilizadores inquiridos.
3. Pesquisar as respostas oferecidas tendo em conta a relevância que um determinado formato tem na profissão do indivíduo inquirido.

## OPÇÕES METODOLÓGICAS

### Tipologia do estudo:

Este estudo é de carácter descritivo e fazendo utilização de inquéritos.

População/amostra:

A população alvo visava os utilizadores de imagem vectorial, em Portugal, mas devido à impossibilidade prática de inquirir toda esta população, foi seleccionada uma amostra por conveniência, constituída por utilizadores que trabalham ou estudam na área do processamento de imagem.

Critério de inclusão:

Pessoas cujo ramo de actividade profissional envolvesse o uso de imagens vectoriais, como operadores CAD, publicitários, entre outros. Foram também inquiridos alunos de Design Multimédia cuja frequência no curso fosse igual ou superior ao 3º ano.

Método/instrumento de recolha de dados:

A recolha de dados foi realizada através de um inquérito constituído por 9 perguntas fechadas, de escala tipo Likert, as correspondências serão mencionadas mais à frente.

As restantes 8 perguntas são de resposta aberta, de forma a conhecer individualmente a relevância da resposta do inquirido assim como o seu conhecimento do tema. (*vide* Anexo I).

## Tratamento de dados:

A sumarização das respostas dadas pelos inquiridos está presente na seguinte tabela.

Propriedades da imagem	Irrelevante	Pouco importante	Relevante	Importante	Vital	Não sabe
<b>Tamanho:</b> Espaço que a imagem ocupa (por exemplo em disco)	2	10	2	16	0	2
<b>Número de utilizadores:</b> Número de pessoas (clientes ou parceiros de trabalho) que utilizam este formato	0	9	2	5	9	7
<b>Tempo de processamento:</b> Velocidade em que a imagem é aberta nos seus programas, velocidade de modificação e velocidade com que grava para ficheiro	0	0	9	17	6	0
<b>Número de programas que suportam:</b> Número de programas que abrem, manipulam e gravam o formato de imagem (utilizados pelos clientes e parceiros)	5	0	1	14	12	0
<b>Programas que suportam:</b> Programa específico que suporta o formato	6	0	2	15	9	0
<b>Empresa que desenvolve:</b> Importância da empresa que lança, actualiza e eventualmente deixa de dar continuação a um formato de imagem	1	5	11	9	6	0
<b>Transferência na Internet:</b> Facilidade e velocidade de transferência da imagem <i>via</i> Internet (para ou de clientes e parceiros)	0	0	10	9	13	0
<b>Quantidade e qualidade de informação sobre o formato:</b> Importância sobre a quantidade de informação disponível para consulta pelo público geral sobre o formato e respectiva qualidade	0	4	11	13	4	0
<b>Formato não proprietário:</b> Importância de ser um formato que não obriga a pagamentos para ser utilizado	0	0	0	12	17	3

Tabela 13: Resultados do Inquérito

## 5.1.2 FASE EMPÍRICA

### ANÁLISE DE DADOS

Para dar uma visão mais precisa dos resultados obtidos, é dada a seguinte cotação, segundo a escala de Likert:

- 1 para as respostas que dão importância “irrelevante”
- 2 para as respostas que dão importância “pouco relevante”
- 3 para as respostas que dão importância “relevante”
- 4 para as respostas que dão importância “importante”
- 5 para as respostas que dão importância “vital”

Em cada propriedade será depois apresentada uma média de valores, podendo-se consultar quais as propriedades a que os inquiridos dão mais relevância. As respostas “não sabe” serão ignoradas, e a média será feita sem ter em conta os elementos que responderam “não sabe” nessa questão. A média é então calculada da seguinte forma:

Seja  $ns$  = número das respostas “Não sabe”.

Seja  $ir$  = número das respostas “Irrelevante”.

Seja  $pi$  = número das respostas “Pouco importante”.

Seja  $re$  = número das respostas “Relevante”.

Seja  $im$  = número das respostas “Importante”.

Seja  $vi$  = número das respostas “Vital”.

$$\text{Média} = \frac{(1 \times ir) + (2 \times pi) + (3 \times re) + (4 \times im) + (5 \times vi)}{32 - ns}$$

$$32 - ns$$

Segue-se a análise das propriedades presentes no inquérito juntamente com as respostas recebidas.

#### TAMANHO

Em relação ao tamanho que a imagem ocupa em disco, é a única propriedade a que ninguém atribuiu a importância vital. Como era previsto antes do estudo, isto pode dever-se ao facto de não ter sido inquirido ninguém com um grande espaço em disco ocupado, ou actividade profissional exigente a esse nível. A maioria das respostas situa esta propriedade como importante, no entanto a média é de 2,88 (pouco inferior a relevante).

#### NÚMERO DE UTILIZADORES

Em relação ao número de utilizadores, é a propriedade com o maior número de respostas “não sabe”. É uma das propriedades em que as opiniões mais se dividem, talvez por se tratar de utilizadores que maioritariamente não trocam informações com outros utilizadores. Seria de esperar que no caso de uma empresa que recebe imagens de clientes e depois tem que as processar, o número de utilizadores fosse uma propriedade prioritária. A média obtida para esta característica é de 3,56 o que faz desta propriedade mais importante que o tamanho que a imagem ocupa em disco.

#### TEMPO DE PROCESSAMENTO

O tempo de processamento obteve o maior número de respostas “importante” e uma das três propriedades com menores números de respostas “irrelevante” e “pouco importante”. Esta seria de esperar uma das características mais relevantes, e tal veio a confirmar-se. A média obtida por esta propriedade é de 3.9, o que a posiciona um pouco abaixo de importante.

#### NÚMERO DE PROGRAMAS QUE SUPORTAM

O número de programas que suportam obteve um conjunto de respostas bastante abrangente, tendo o segundo maior número de respostas “irrelevante” e o terceiro maior número de respostas “vital”. A média é de 3.88.

#### PROGRAMAS QUE SUPORTAM

Programa específico que suporta o formato é uma propriedade que também divide as opiniões. A média é de 3.7 aproximadamente.

#### EMPRESA QUE DESENVOLVE

A empresa que desenvolve obteve a grande maioria das respostas no “relevante”. A média é de 3.4.

#### TRANSFERÊNCIA NA INTERNET

A transferência na internet obteve um conjunto de respostas sem nenhuma “irrelevante” ou “pouco relevante”. Isto é de estranhar, visto a taxa de transferência estar directamente relacionada com o tamanho que a imagem ocupa. A média é de 4.

#### QUANTIDADE E QUALIDADE DE INFORMAÇÃO SOBRE O FORMATO

A quantidade e qualidade de informação sobre o formato também surpreende pelo facto desta informação interessar principalmente a programadores e *developers*. Os inquiridos mostraram uma sensibilidade em relação a esta propriedade superior à esperada antes do estudo. A média é de 3.5.

#### FORMATO NÃO PROPRIETÁRIO

Por fim e para maior surpresa, os inquiridos mostraram que o facto de um formato ser ou não proprietário tem uma importância bastante acima de todas as

outras propriedades. O facto de um formato ser não proprietário é bastante mais importante que o espaço que um determinado formato ocupa em disco. Esta propriedade não obteve respostas “irrelevante”, “pouco relevante” ou até “relevante”. A média é de 4.6 aproximadamente.

## DISCUSSÃO DOS RESULTADOS

Os resultados apontam para a necessidade de um estudo mais abrangente e aprofundado, pois verificou-se uma necessidade de dar mais importância a esta análise.

Todos os documentos estudados sobre escolhas de formatos, para além de não terem em conta as preferências dos utilizadores, tendem a ter como factor importante de preferência, o espaço que a imagem ocupa em disco.

Factores como o tempo de processamento, para além de relativos são difíceis de apurar. No entanto este estudo mostra que os factores pelos quais outros programadores se guiam, podem ser na realidade dos menos importantes para os utilizadores.

Assim, por ordem decrescente de importância, este estudo indica que devemos ter em conta:

1. Se um formato é ou não proprietário
2. Velocidade e facilidade de transferência na internet
3. Tempo de processamento
4. Número de programas que suportam o formato
5. Programa específico que suporta o formato (depende da área de trabalho)
6. Número de utilizadores
7. Quantidade e qualidade da informação sobre o formato
8. Empresa que desenvolve
9. Tamanho que a imagem ocupa em disco

É de realçar as dificuldades inerentes a este estudo, sendo que os profissionais desta área, nem sempre são fáceis de contactar, e nem sempre se mostram participativos.

Concretamente, foram distribuídos 92 inquéritos, tendo sido obtidas respostas em apenas 39 dos casos. Dos inquéritos reunidos 7 foram excluídos do estudo devido ao insuficiente conhecimento da área por parte dos inquiridos. Também a selecção de pessoas com conhecimento na área nem sempre se revelou uma tarefa fácil.

As perguntas de resposta aberta tinham objectivos diferentes. O primeiro grupo de perguntas com resposta aberta foi projectado para pesar o valor das respostas, no entanto, devido ao não preenchimento pela maioria dos indivíduos esta hipótese foi anulada.

No segundo grupo de respostas abertas, onde se pergunta quais as preferências de formatos e softwares, foram excluídos os indivíduos cujas respostas fossem insatisfatórias, ou seja, respostas referentes a formatos não vectoriais ou programas de edição gráfica não vectorial.

No ultimo grupo, onde se perguntava qual a profissão do individuo inquirido, a resposta serviu apenas para certificar que todos os indivíduos inquiridos estavam englobados pelo critério de inclusão. Esta última questão pretendia ser um mecanismo de controlo adicional, não se tendo revelado contudo necessária, uma vez que a entrega em mão do questionário a cada um dos indivíduos permitiu garantir que todos eram elegíveis para amostra.

## 5.2 DrawIt – Descrição e manual de utilizador

Para concluir o estudo sobre imagens vectoriais, foi construído um editor de imagem ao qual foi dado o nome de DrawIt.

Este sistema suporta funções para introdução e edição de texto e formas geométricas, possibilitando escalonamentos, translações, rotações e controlo de cor.

Este software possui como formato nativo o SVG, estudado ao longo da tese.

A elaboração deste sistema, envolveu um estudo aprofundado da norma SVG 1.1, e permitiu a utilização das imagens vectoriais sob o ponto de vista do programador.

Esta foi de longe a parcela mais morosa, trabalhosa e enriquecedora de todo este trabalho. A ideia para o DrawIt nasceu acerca de dois anos. Esta versão, consiste numa reformulação completa do sistema desenvolvido anteriormente, partilhando com as anteriores apenas os objectivos a que se propõe. A ideia inicial para este software englobava também a possibilidade de este controlar uma plotter para que os desenhos realizados no software pudessem ser reproduzidos noutros suportes (papel, madeira...).

A criação deste software envolveu uma análise das ferramentas disponíveis e das suas capacidades, assim como os benefícios que poderiam trazer ao projecto. Assim, após esta fase inicial, o código do projecto foi migrado de Delphi 6 para Java produzido através do NetBeans 6.0.1, muito em parte devido ao apoio para DOM's e interface com código XML.

As ferramentas utilizadas possibilitam que o DrawIt seja um pouco mais que um mero editor de imagem, visto que possibilita:

- Edição de imagem: Capacidade de processamento gráfico com as propriedades descritas posteriormente implementadas. O acréscimo de mais funcionalidades é apenas dificultado pelo tempo necessário, visto que o seu suporte está garantido.
- Controlo de Hardware: O DrawIt permite o controlo de uma plotter desenvolvida no âmbito do projecto de final de Licenciatura. Esta plotter tem como objectivo desenhar a imagem do editada pelo DrawIt em suporte de papel. Assim sendo, e devido à componente de Hardware não ter sido alterada, esta capacidade não irá ser abordada no seguimento do presente documento.

No entanto, para o software continuar a permitir esta propriedade, mecanismos de discretização dos elementos gráficos, e métodos de comunicação tiveram que ser implementados.

- Controlos por voz: O DrawIt permite controlo de algumas funcionalidades por voz, tais como abrir ficheiros, e parar o processamento. Parar o processamento torna-se principalmente relevante quando temos em conta que o DrawIt controla Hardware, e como tal, interrompe as comunicações.
- Vectorização: Através da ferramenta de nome autotrace, é permitido ao DrawIt criar ficheiros SVG a partir de ficheiros JPG e PNG.
- Rasterização: Através do Batik, o DrawIt permite rasterizar gráficos SVG.

O novo DrawIt acrescenta uma grande melhoria na velocidade de processamento em relação à versão anterior. Isto deve-se a que todas as funções de processamento mais exigentes são agora realizadas em threads independentes, não interferindo directamente com o processamento do processo principal. Esta característica é imperceptível graficamente mas exigiu bastante tempo, e obteve como consequência, um ganho na velocidade de processamento de cerca de 60%. Este valor obtido no ganho, resulta das medições do tempo de processamento de imagens idênticas no sistema sem threads e no sistema com threads.

Segue-se então a discriminação das capacidades e modo de funcionamento do DrawIt.

Devido à possível dificuldade de interpretação da imagem por parte do leitor, foram criadas áreas que iram ser abordadas detalhadamente no seguimento do documento. Assim, foram criadas 6 áreas referentes às seguintes parcelas da imagem:

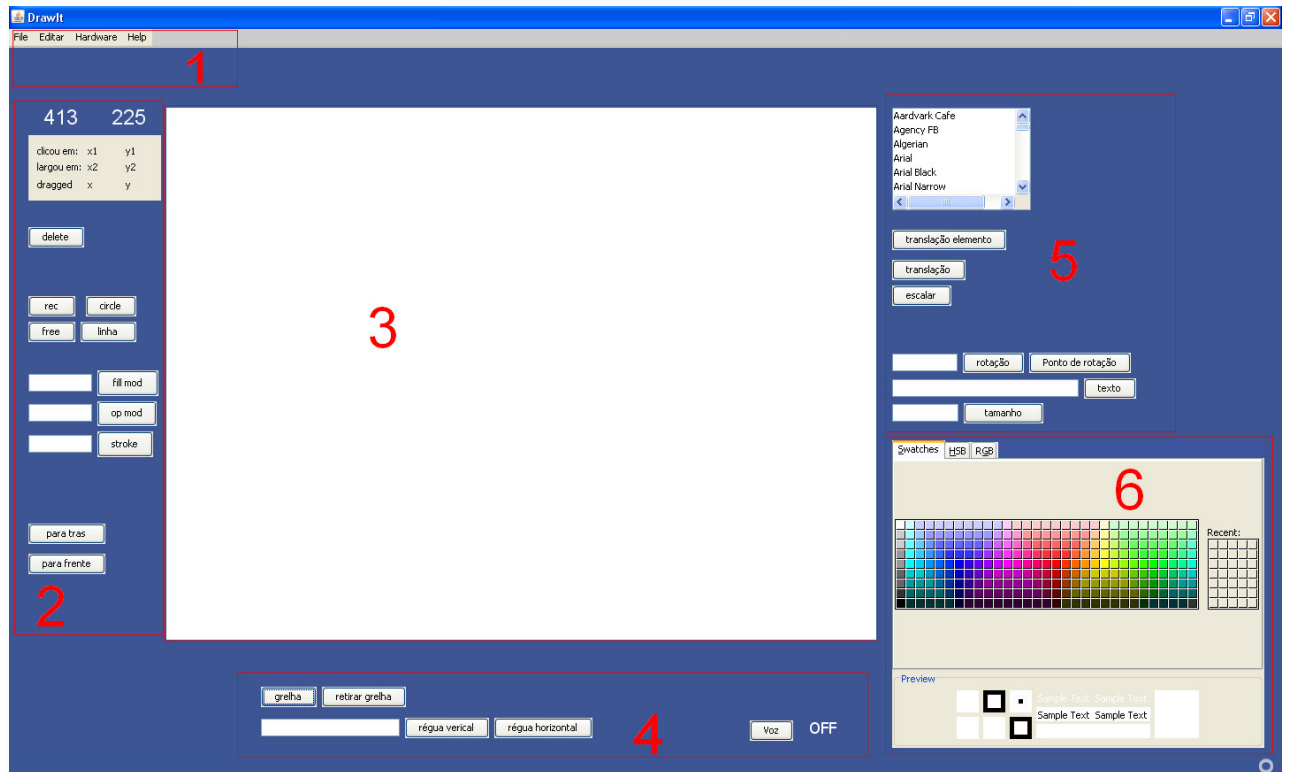


Figura 9: Representação das 6 áreas do interface principal

Mostrando em pormenor o mecanismo de interface com o utilizador, temos na parte superior da disposição (área 1), o conjunto de menus que se segue:

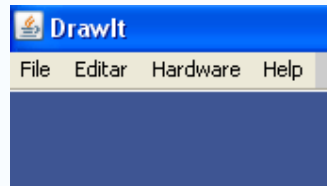


Figura 10: Representação da área 1

Dentro do menu “File”, temos as seguintes possibilidades:

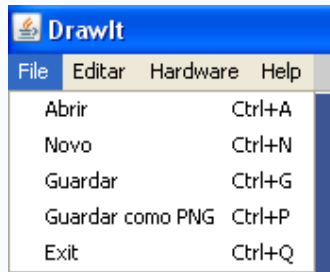


Figura 11: Menu “File”

Escolhendo a opção “Abrir”, podemos abrir um ficheiro SVG da nossa escolha. Esta escolha é feita através do seguinte interface:

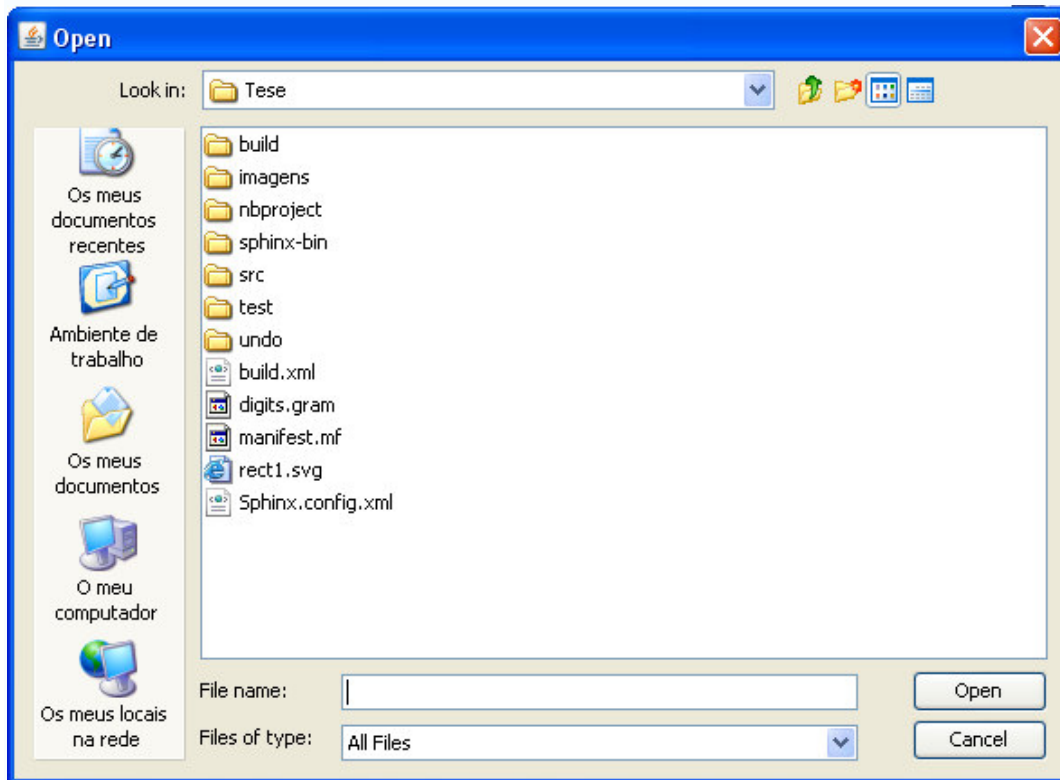


Figura 12: Interface de abertura de ficheiros

Escolhendo a opção “Novo”, podemos criar um novo documento SVG directamente na área de trabalho do DrawIt (área 3).

Pressionando a opção “Guardar”, podemos salvar a imagem mostrada na área de trabalho, podendo escolher um ficheiro de destino.

Esta escolha é feita através da seguinte interface:

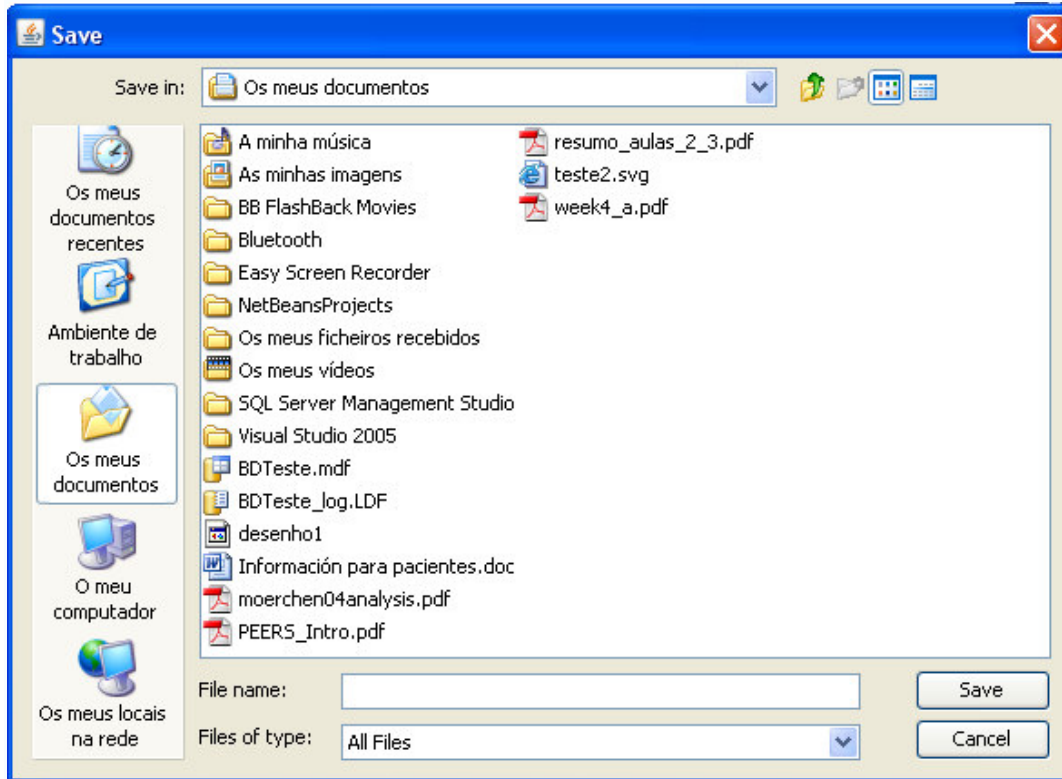


Figura 13: Interface de guardar ficheiros no formato SVG

Premindo a opção “Guardar” como PNG, o ficheiro SVG é rasterizado e guardado num ficheiro com o mesmo nome, mas com o formato PNG.

A opção “Exit” permite sair do programa.

No menu “Editar”, temos as seguintes funcionalidades:



Figura 14: Menu “Editar”

Pressionando a opção “Undo”, desfazemos a última alteração feita à imagem, sempre que aplicável.

Na opção “Redo”, desfazemos o efeito da última operação de “Undo”, sempre que aplicável.

No Menu “Hardware”, temos a seguinte opção:



Figura 15: Menu “Harware”

Pressionando a opção “Menu Hardware”, é-nos disponibilizada o seguinte conjunto de botões, sob a forma de um novo painel:

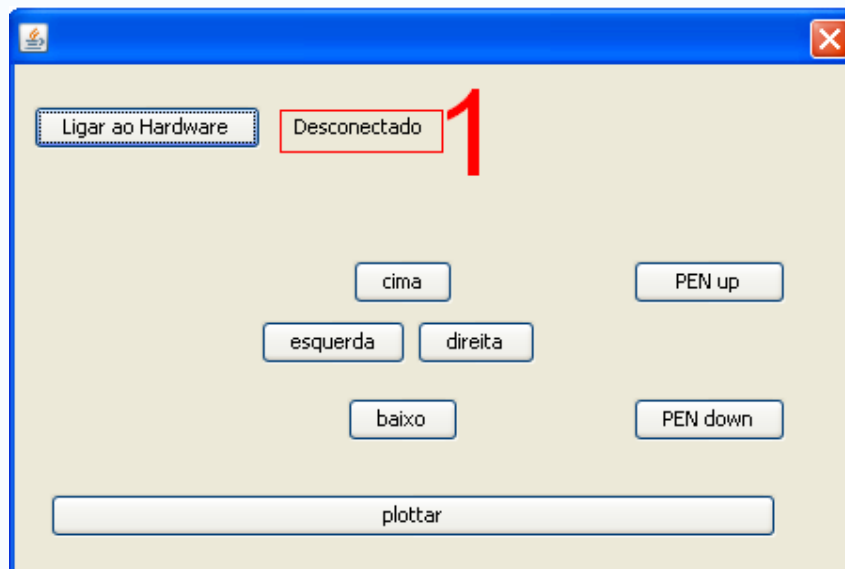


Figura 16: Interface de controlo de Hardware

O botão “Ligar ao Hardware” permite no caso da *label* (assinalada com o número 1) à sua direita mostrar a palavra Desconectado, ligar à placa de interface com o Hardware.

Uma vez conectado, a *label* passará a exibir a palavra Conectado.

Existem quatro botões direccionais que permitem movimentar a plotter no sentido pretendido. Esses botões exibem as palavras “cima”, “esquerda”, “baixo”, e “direita”, sendo auto-explicativos.

O botão “PEN up” permite a deslocação da cabeça da plotter no sentido ascendente, enquanto o botão “PEN down” movimenta a cabeça no sentido descendente.

O botão “plottar” permite que a imagem seja processada, discretizando os elementos gráficos e comunicando as referências de cada ponto extraído para o Hardware.

No menu “Help” temos a seguinte opção:

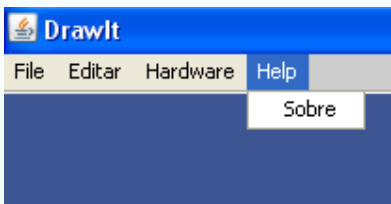


Figura 17: Menu “Help”

Pressionando a opção “sobre”, obtemos uma janela, com as informações básicas do software e dos seus criadores, que é ilustrada abaixo.



Figura 18: Interface de informação do software

Referentes à área 2 temos as seguintes funções:



Figura 19: Área 2

As duas primeiras *labels* que exibem os valores 287 e 131 (assinaladas com o número 1), dizem respeito à posição actual do ponteiro do rato na área de trabalho, referindo o eixo-x e eixo-y respectivamente.

O botão “delete”, situado abaixo do menu anterior permite apagar o elemento gráfico seleccionado.

O botão “rec”, permite a criação de um rectângulo da próxima vez que o utilizador clicar na área de trabalho. Da mesma forma, o botão “circle” permite desenhar um círculo na área de trabalho. O botão “free” permite desenhar livremente enquanto se pressiona o botão esquerdo do rato na área de trabalho. O botão “linha” permite a criação de linhas rectas.

Mais abaixo surgem três botões associados a caixas de diálogo.

O primeiro, o botão “fill mod”, permite modificar o atributo “fill” (pintar o interior dos elementos), tendo em conta a cor seleccionada na *pallette* que iremos ver mais adiante(área 6), ou então tendo em conta a cor referida na caixa de texto correspondente.

O botão “op mod” permite alterar a opacidade do elemento gráfico seleccionado tendo em conta o valor colocado na caixa de diálogo correspondente.

O botão “stroke” funciona da mesma forma que o botão “fill mod”, mas permite alterar a cor das linhas que definem o objecto seleccionado.

O botão “para trás” permite recuar o elemento gráfico na hierarquia do SVG, obtendo o efeito de *layering*. Assim, este botão coloca o elemento na *layer* anterior à sua *layer* actual.

Da mesma forma o botão “para frente” permite colocar o elemento numa *layer* superior à actual.

No centro do sistema temos a área de trabalho (área 3) como se mostra na seguinte figura:

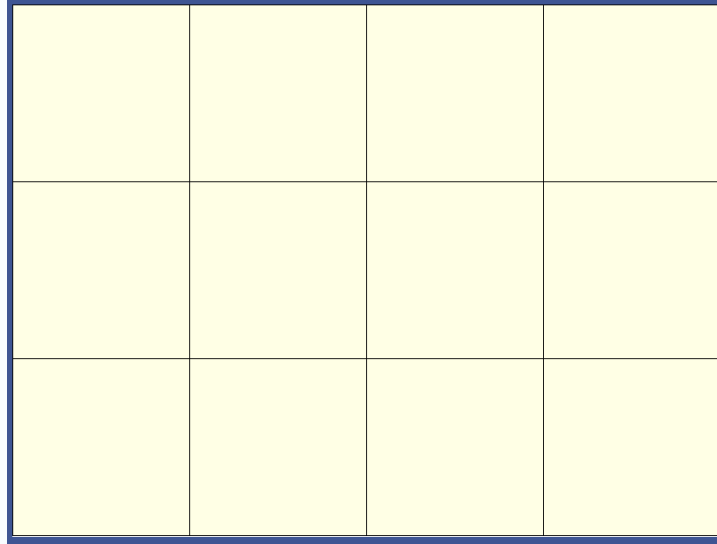


Figura 20: Área 3

A área de trabalho está inicialmente totalmente branca, mas no entanto foram aqui aplicadas grelhas para a melhor percepção do leitor.

Tem um tamanho de 800 pixéis de largura por 600 de altura, é a parte onde são *renderizadas* as imagens SVG, e no qual é permitida a realização de algumas modificações.

Por baixo da área de trabalho (área 4) temos os seguintes botões:

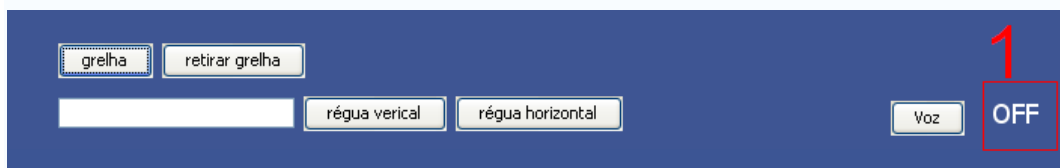


Figura 21: Área 4

O botão “grelha” insere um conjunto de linhas em forma de grelha na área de trabalho como mecanismo de referência. O botão “retirar grelha” retira qualquer mecanismo regulador na área de trabalho.

Os botões “régua vertical” e “régua horizontal” permitem o desenho de uma linha reguladora tendo em conta a coordenada especificada na caixa de texto, na orientação correspondente ao botão seleccionado.

O botão “Voz”, permite ligar ou desligar o interface com o microfone, sendo o seu estado activo assinalado pela palavra ON na *label* da direita(assinalada com o número 1), e o estado activo assinalado pela palavra OFF na mesma *label*.

Na parte direita à área de trabalho (área 5) temos a seguinte interface:

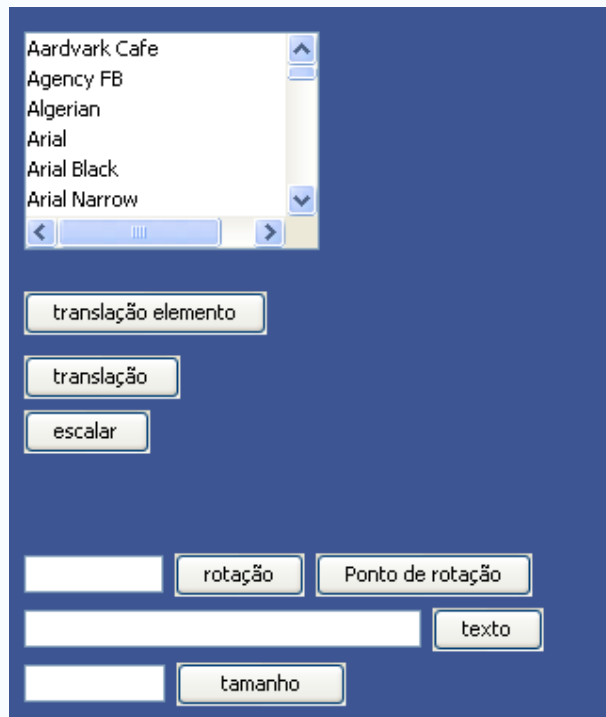


Figura 22: Área 5

A lista que aparece no topo, enumera as *fonts* disponíveis na máquina, pelo que é através desta função que se pode alterar o tipo de letra do texto.

O botão “translação elemento”, permite que o próximo arrastamento na área de trabalho desencadeie uma translação no elemento seleccionado no momento do click no botão.

O botão “translação”, quando seleccionado, permite que o próximo drag na área de trabalho realize uma translação de todos os elementos gráficos da imagem.

O botão “escalar” permite um aumento ou diminuição do tamanho de todos os elementos gráficos da imagem tendo em conta o próximo drag na área de trabalho.

Clicando no botão “rotação” faz com que o objecto seleccionado sofra uma rotação num ângulo descrito no campo de texto à sua esquerda.

O botão “Ponto de rotação” permite criar um ponto na área de trabalho sobre o qual vai ser exercida a rotação do elemento seleccionado.

O botão “texto”, adiciona no próximo click na área de trabalho, um texto proveniente do campo de texto à sua esquerda.

O botão “tamanho” altera o tamanho do texto para o valor contido na caixa de texto à sua esquerda.

Por fim, temos a área de selecção de cor (área 6) que contém três componentes que permitem a selecção da cor de diferentes formas. O primeiro componente é ilustrado abaixo.

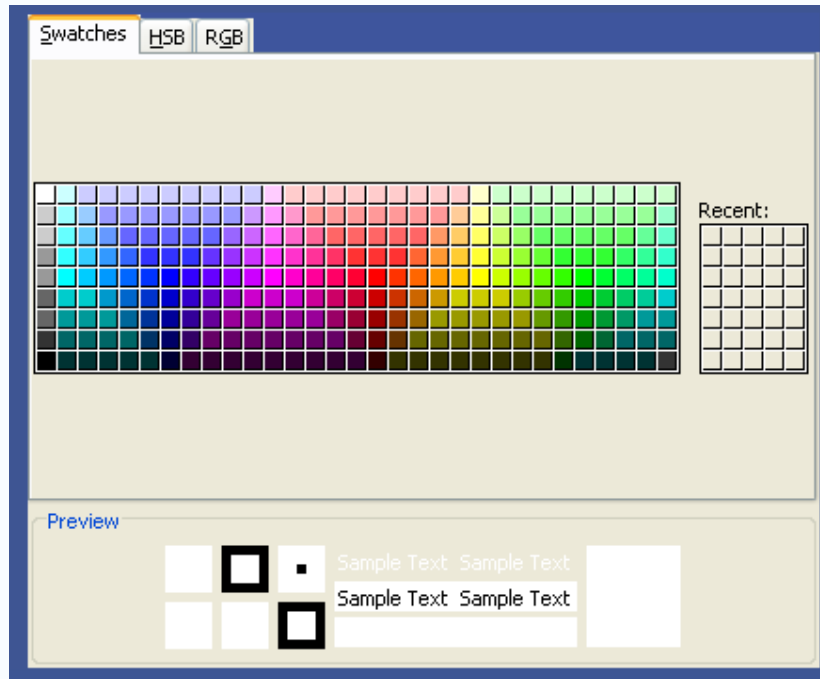


Figura 23: Área 6, componente por omissão

Este componente permite a selecção de cores, disponibilizando uma pré-visualização na área de “preview”, e um historial das cores seleccionadas à esquerda.

O segundo componente é o seguinte:

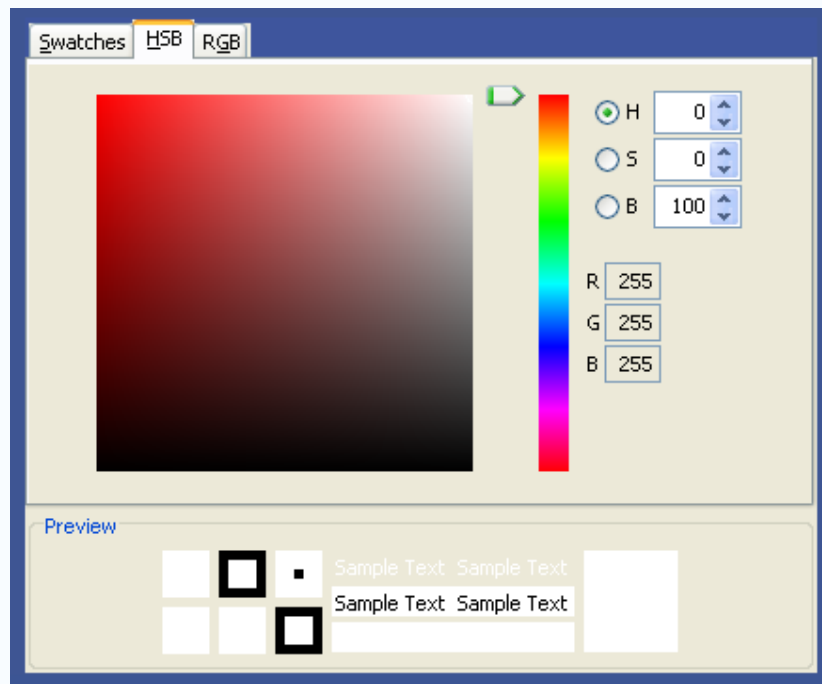


Figura 24: Área 6, segundo componente

Permite a selecção da cor através do clicar do botão na área colorida, e através da selecção de valores do método HSB (hue, saturation, e brightness).

Da mesma forma uma pré-visualização está disponível na parte inferior.

O terceiro componente é o seguinte:

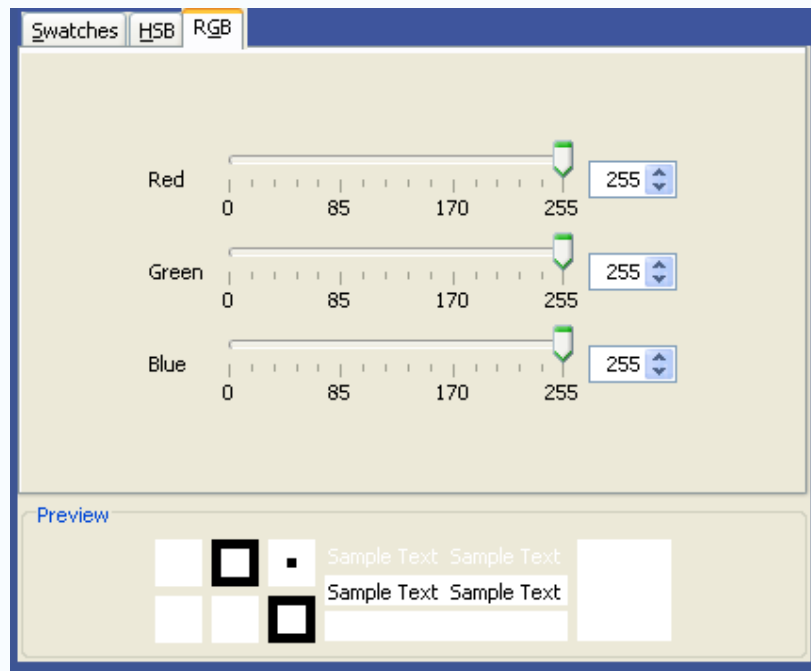


Figura 25: Área 6, terceiro componente

Permite a selecção de cores através de valores RGB (Red, Green e Blue).

Todas as funções descritas neste manual de utilização, alteram a forma de *renderização* da imagem através de modificações dos seus elementos. No entanto existem também formas de navegação, que alteram a forma de visualização da imagem, mas não realizam nenhuma alteração nos seus componentes.

Estes métodos modificam apenas os mecanismos de *renderização*, e são eles:

1. *Zoom*
2. *Zoom* por selecção de área
3. Rotação
4. Translação

Por fim podemos ver o sistema com um gráfico *renderizado* na área de trabalho para se poder perceber melhor o modo de funcionamento do DrawIt.

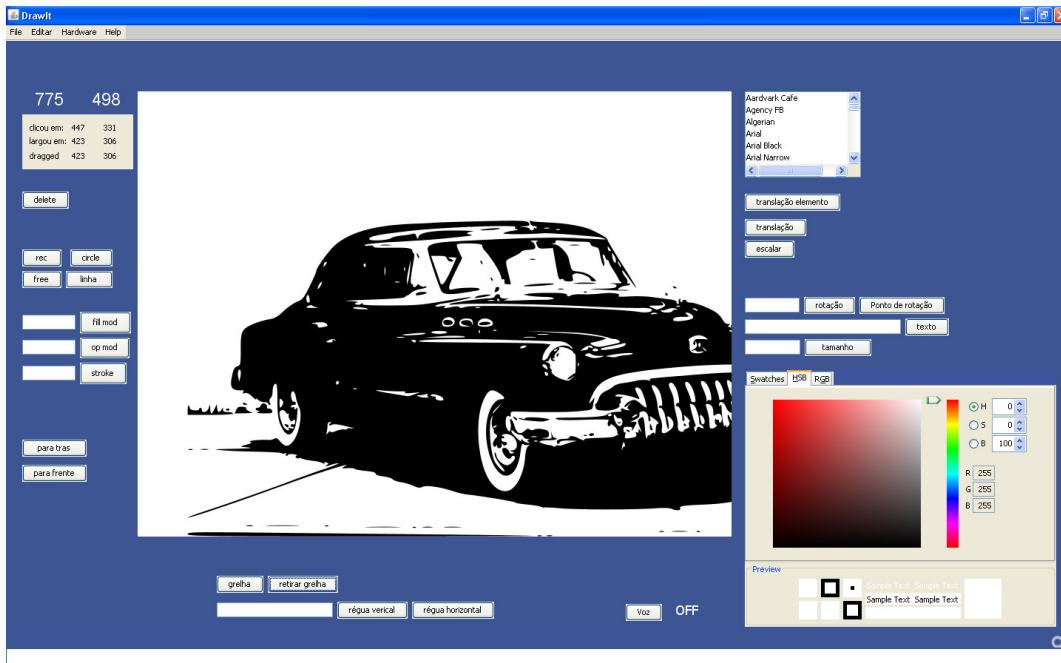
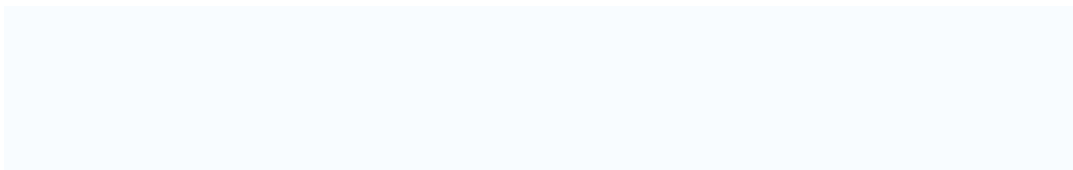


Figura 26: Visão geral do DrawIT com imagem renderizada



## **6. PERSPECTIVAS FUTURAS**

---

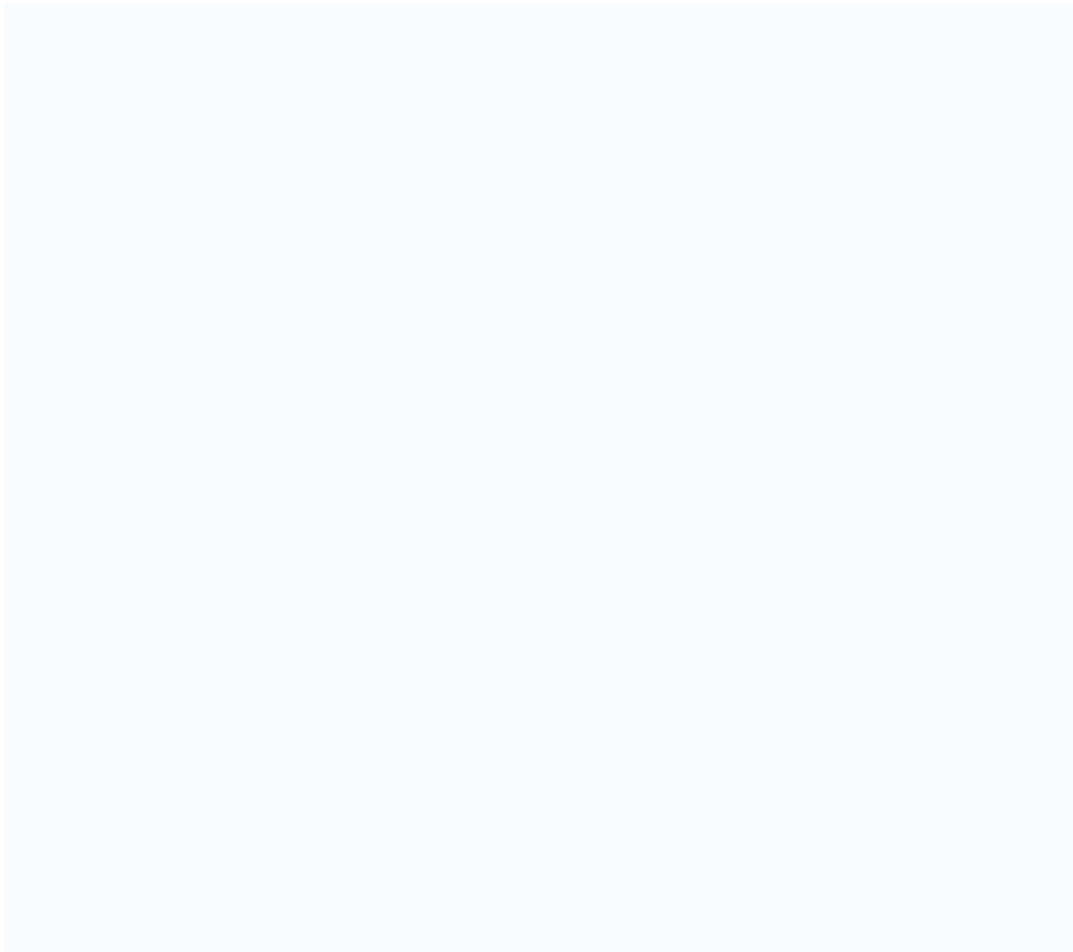


Devido ao facto de qualquer programa de edição gráfica estar sempre longe de atingir todas as suas potencialidades, o DrawIt pode tornar-se ainda mais poderoso. As capacidades actuais do software implicaram a construção de uma base complexa que poderá suportar um conjunto de funcionalidades adicionais.

Em relação ao controlo por voz, este poderia ser mais completo, mas exigiria provavelmente uma mudança na ferramenta utilizada deixando para trás o Sphinx 4.

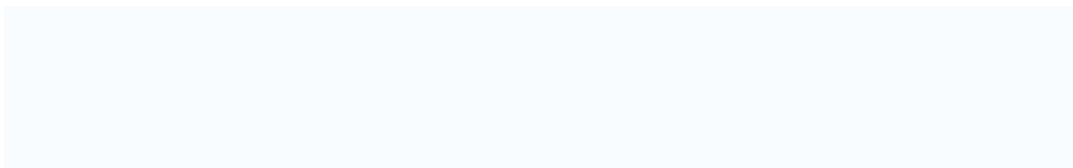
Com as devidas condições poderá levar-se a cabo um estudo mais aprofundado e abrangente das preferências dos utilizadores de imagens vectoriais. Isto traria uma possível melhoria na qualidade dos softwares de imagem, uma vez que o formato escolhido representa uma parte fundamental do seu modo de funcionamento.

Em relação às imagens vectoriais, prevê-se um aumento na sua utilização, juntamente com o aumento do seu suporte pelos sistemas informáticos, o que vem fundamentar e justificar o estudo realizado<sup>(41)</sup>.



## **7. CONCLUSÕES**

---



Relativamente às imagens vectoriais, existe bastante informação de qualidade sobre a sua utilização, propósito e comparação com imagens *raster*.

Sobre os formatos existentes deste tipo de imagens, está disponível menos informação do que a esperada. Foi notória a existência de muitos formatos, dispersando com eles os utilizadores e programadores.

Foram consultados estudos<sup>(1)</sup>, que referiam a necessidade de *standards* poderosos para este ramo da tecnologia. No final deste estudo, considero também essa mesma necessidade.

Relativamente aos métodos de armazenamento de longa duração, conclui-se que os formatos escolhidos devem ser um dos seguintes: SVG, PDF/A e WebCGM. Estes formatos possuem características que permitem ou facilitam a manutenção da integridade da informação no intercâmbio entre plataformas.

O estudo das preferências dos utilizadores revelou que é possível que alguns dos factores pelos quais os programadores se guiam na construção dos seus softwares possam estar errados. Conclui-se que existe uma necessidade de estudo nesta área.

No que toca ao DrawIt, este possibilitou o estudo aprofundado da norma SVG 1.1. Do estudo realizado sobre este formato, conclui-se que está ainda numa fase de crescimento, sendo que a norma SVG 1.2 está já sob construção.

As ferramentas utilizadas facilitam em grande parte o processamento de imagens vectoriais, sendo no entanto que a informação referente às mesmas é por vezes insuficiente.

A norma SVG 1.1 é bastante extensa e a sua capacidade de descrever funções através de várias formas distintas aumenta o esforço do programador.

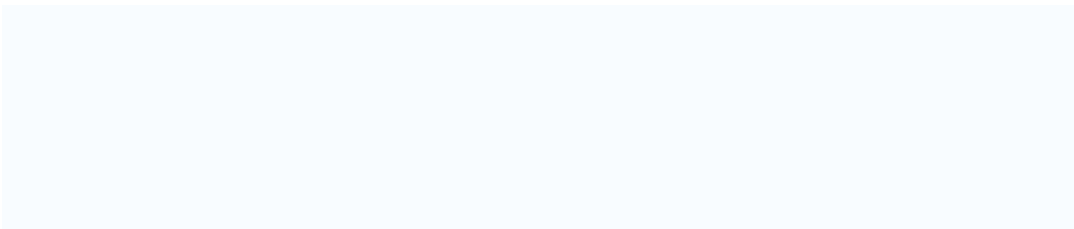
A migração de Delphi 6 para Netbeans 6.0.1, foi na sua generalidade uma boa opção tendo no entanto que apontar as seguintes conclusões:

- A facilidade de construção de interfaces em Delphi é significativamente mais fácil e rápida que em Netbeans
- Da experiência anterior com o Windows Speech API, o Sphinx 4 fica bastante aquém no que toca à facilidade de incorporação no código, simplicidade dos ficheiros de configuração e informação disponível
- Existe um número muito superior de utilizadores e ferramentas Java relacionadas com SVG e XML
- O Batik revelou-se uma ferramenta muito útil apesar da informação disponível sobre a sua utilização nem sempre ser clara e relevante.



## **BIBLIOGRAFIA**

---



1. Coyne M., Duce D., Hopgood B., Mallen G. and Stapleton M., 2007, *JISC Digital Preservation Programme: Study on the Significant Properties of Vector Images, versão 4.3*. Disponível na World Wide Web em [http://www.jisc.ac.uk/media/documents/programmes/preservation/vector\\_images.pdf](http://www.jisc.ac.uk/media/documents/programmes/preservation/vector_images.pdf) (Consultado em 15 de Março de 2008)
2. Herren T., 2002, *3dc*. Disponível na World Wide Web em <http://on-the-web.ch/3dc/> (Consultado em 21 de Janeiro de 2008)
3. Fercoq R. and van Velsen M., 1997, *3D-Studio File Format (.3ds)*. Disponível na World Wide Web em <http://www.martinreddy.net/gfx/3d/3DS.spec> (Consultado em 13 de Fevereiro de 2008)
4. Adobe Systems Incorporated, 1998, *Adobe Illustrator File Format Specification*. Disponível na World Wide Web em <http://mercator.elte.hu/~saman/hu/okt/AI7FileFormat.pdf> (Consultado em 21 de Janeiro de 2008)
5. BRL-CAD, 2008, *BRL CAD, open source solid modeling*. Disponível na World Wide Web em <http://brlcad.org/> (Consultado em 13 de Fevereiro de 2008)
6. Planetquake, *BSP technical details for Dummies*. Disponível na World Wide Web em <http://qxx.planetquake.gamespy.com/bsp/> (Consultado em 10 de Fevereiro de 2008)
7. ISO 8632, 1998, *STANDARDS for CGM*. Disponível na World Wide Web em [http://www.itl.nist.gov/div897/ctg/graphics/cgm\\_std.htm](http://www.itl.nist.gov/div897/ctg/graphics/cgm_std.htm) (Consultado em 23 de Janeiro de 2008)
8. KHRONOS Group, 2008, *Collada – Digital Asset Exchange Schema for Interactive 3D*. Disponível na World Wide Web em <http://www.khronos.org/collada/> (Consultado em 3 de Fevereiro de 2008)

9. SGI, 2008, OpenGL Overview. Disponível na World Wide Web em <http://www.sgi.com/products/software/opengl/overview.html> (Consultado em 7 de Fevereiro de 2008)
10. Bentley Systems, Incorporated, 2008, OpenDGN Initiative. Disponível na World Wide Web em <http://www.bentley.com/en-US/Products/MicroStation/OpenDGN/> (Consultado em 17 de Fevereiro de 2008)
11. Autodesk, Inc., 2008, Developer Tools. Disponível na World Wide Web em <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5801732>  
(Consultado em 20 de Janeiro de 2008)
12. Computer Knowledge 2008, File Extension .DWG details. Disponível na World Wide Web em <http://on-the-web.ch/3dc/> (Consultado em 22 de Janeiro de 2008)
13. Autodesk, Inc., 2008, DXF reference. Disponível na World Wide Web em <http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=8446698> (Consultado em 6 de Fevereiro de 2008)
14. Adobe Systems Incorporated, 1992, Encapsulated PostScript File Format Specification. Disponível na World Wide Web em [http://partners.adobe.com/public/developer/en/ps/5002.EPSF\\_Spec.pdf](http://partners.adobe.com/public/developer/en/ps/5002.EPSF_Spec.pdf) (Consultado em 14 de Fevereiro de 2008)
15. Sutanthavibul S., 1994, FIG : Facility for Interactive Generation of figures. Disponível na World Wide Web em <http://www.martinreddy.net/gfx/2d/FIG.txt> (Consultado em 2 de Março de 2008)
16. O'Reilly , 2002, GEM VDI. Disponível na World Wide Web em <http://www.fileformat.info/format/gemvector/egff.htm> (Consultado em 7 de Janeiro de 2008)

17. *Artwork Conversion Software, Inc, 1991, D-codes, Apertures and Gerber Plot Files. Disponível na World Wide Web em <http://www.artwork.com/gerber/appl2.htm> (Consultado em 15 de Fevereiro de 2008)*
18. *Understanding HPGL, 1996, Understanding HPGL. Disponível na World Wide Web em <http://cstep.luberth.com/hppl.htm> (Consultado em 1 de Março de 2008)*
19. *U. S. Product Data Association, 1997, Initial Graphics Exchange Specification 5.3. Disponível na World Wide Web em [http://www.uspro.org/documents/IGES5-3\\_forDownload.pdf](http://www.uspro.org/documents/IGES5-3_forDownload.pdf) (Consultado em 19 de Fevereiro de 2008)*
20. *Wavefront and Java3D .obj Format. Disponível na World Wide Web em [http://www.eg-models.de/formats/Format\\_Obj.html](http://www.eg-models.de/formats/Format_Obj.html) (Consultado em 2 de Março de 2008)*
21. *Adobe Systems Incorporated, 2008, PDF Reference. Disponível na World Wide Web em [http://www.adobe.com/devnet/pdf/pdf\\_reference.html](http://www.adobe.com/devnet/pdf/pdf_reference.html) (Consultado em 6 de Janeiro de 2008)*
22. *PDF/A Competence Center, 2008, PDF/A Format. Disponível na World Wide Web em <http://www.pdfa.org/doku.php> (Consultado em 10 de Janeiro de 2008)*
23. *Apple Inc., 1996, Appendix A - Picture Opcodes. Disponível na World Wide Web em <http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-458.html> (Consultado em 5 de Março de 2008)*
24. *Adobe Systems Incorporated, 2008, PostScript vs. PDF: Why do we offer two printing technologies? How do they differ?. Disponível na World Wide Web em <http://www.adobe.com/print/features/psvspdf/index.html> (Consultado em 10 de Fevereiro de 2008)*

25. Bible Software Industry Standards Group, 1998, *The Official STEP Specification, Version 1.1*. Disponível na World Wide Web em <http://www.crosswire.org/bsisg/download.htm> (Consultado em 7 de Março de 2008)
26. 3 Specification for the Simple Vector Format (SVF) v1.1dc. Disponível na World Wide Web em <http://www.softsource.com/svf/spec.html> (Consultado em 23 de Janeiro de 2008)
27. Lilley C. and Schepers D., 2008, *Scalable Vector Graphics (SVG) XML Graphics for the Web*. Disponível na World Wide Web em <http://www.w3.org/Graphics/SVG/> (Consultado em 5 de Janeiro de 2008)
28. Adobe Systems Incorporated, 2008, *SWF File Format Specification, Version 9*. Disponível na World Wide Web em [http://www.adobe.com/devnet/swf/pdf/swf\\_file\\_format\\_spec\\_v9.pdf](http://www.adobe.com/devnet/swf/pdf/swf_file_format_spec_v9.pdf) (Consultado em 16 de Fevereiro de 2008)
29. Andersson O., Armstrong P., Axelsson H. et al., 2006, *Scalable Vector Graphics (SVG) Tiny 1.2 Specification*. Disponível na World Wide Web em <http://www.w3.org/TR/SVGMobile12/> (Consultado em 5 de Janeiro de 2008)
30. ECMA-363, 2007, *Standard ECMA-363, Universal 3D File Format*. Disponível na World Wide Web em <http://www.ecma-international.org/publications/standards/Ecma-363.htm> (Consultado em 7 de Março de 2008)
31. Microsoft Corporation, 1998, *Vector Markup Language (VML)*. Disponível na World Wide Web em <http://www.w3.org/TR/NOTE-VML> (Consultado em 10 de Março de 2008)
32. Microsoft Corporation, 2007, *Windows Metafile Format (wmf) Specification*. Disponível na World Wide Web em [http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/WindowsMetafileFormat\(wmf\)Specification.pdf](http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/WindowsMetafileFormat(wmf)Specification.pdf) (Consultado em 7 de Fevereiro de 2008)

33. Microsoft Corporation, 2008, *Enhanced Metafile Format Specification*. Disponível na World Wide Web em <http://msdn2.microsoft.com/en-us/library/cc204166.aspx> (Consultado em 10 de Fevereiro de 2008)
34. Almeida, L. S., 2003, *Metodologia da investigação em psicologia e educação*. Braga, Lusografe, 3ª edição
35. Fortin, M., 1999, *O processo de investigação*. Loures, Lusociência
36. Ribeiro, J., 2007, *Metodologia de investigação em psicologia e saúde*. Porto, Legis Editora
37. The Apache Software Foundation, 2008, *Batik Java SVG Toolkit*. Disponível na World Wide Web em <http://xmlgraphics.apache.org/batik/> (Consultado em 5 de Janeiro de 2008)
38. Carnegie Mellon University, 2004, *Sphinx-4 A speech recognizer written entirely in the Java™ programming language*. Disponível na World Wide Web em <http://cmusphinx.sourceforge.net/sphinx4/> (Consultado em 17 de Fevereiro de 2008)
39. Martin Weber, 2004, *AutoTrace*. Disponível na World Wide Web em <http://autotrace.sourceforge.net/> (Consultado em 15 de Janeiro de 2008)
40. Mike Doughty, 2007, *Two Kinds of Computer Graphics*. Disponível na World Wide Web em <http://www.sketchpad.net/basics1.htm> (Consultado em 7 de Janeiro de 2008)
41. SVG.org, 2008, *the future of web graphics*. Disponível na World Wide Web em <http://svg.org/story/2008/4/7/155544/8359> (Consultado em 29 de Janeiro de 2008)

**ANEXOS**

---



**ANEXO I**

---

# Inquérito sobre imagens Vectoriais.

## **Nota introdutória:**

Existem no mercado diversos formatos de imagem vectorial, sendo as suas características conhecidas, assim como estudos teóricos e práticos sobre as diferenças entre eles.

No entanto, quando se procura saber qual o formato de imagem a utilizar na criação de um dado software, um estudo sobre as necessidades dos utilizadores é fundamental.

Este inquérito surge devido à falta de informação sobre quais as propriedades que utilizadores de imagens vectoriais dão mais relevância num formato de imagem.

Se lhe foi pedido o preenchimento deste documento, lembre-se que o produto final deste trabalho será para o seu benefício, pelo que deverá responder da forma mais precisa que lhe for possível.

## Informações sobre o preenchimento:

O inquérito está dividido em quatro partes, representadas por quatro tabelas distintas:

**Propriedades da imagem**

**Caracterização do trabalho**

**Informação sobre preferências**

**Identificação do trabalho**

Deverá fazer corresponder a todos os campos das **propriedades de imagem** apenas uma nota dependendo da importância que aquela propriedade tem para o seu trabalho.

Esta nota será marcada com um X na respectiva coluna.

A nota que deverá dar, deverá ser uma das seguintes:

**Irrelevante:** que não tem qualquer significado para o seu trabalho.

**Pouco importante:** que não tem impacto significativo no seu trabalho.

**Relevante:** que tem impacto significativo no seu trabalho.

**Importante:** que tem um impacto muito significativo e frequente no seu trabalho.

**Vital:** que é fundamental no seu trabalho. Uma imagem que tenha esta característica pouco eficiente representa uma dificuldade ou barreira no seu trabalho.

**Não Sabe:** Não tem informação suficiente sobre essa característica.

Na parte de **caracterização do seu trabalho**, deverá responder com valores numéricos maiores ou iguais que 0, acompanhados da respectiva unidade de medida.

Nestes campos, é pedida uma estimativa. Caso não possua qualquer informação sobre este tipo de quantidades coloque a palavra “desconheço” nesse determinado campo.

Nos campos de **Informação sobre preferências e nos campos de Identificação do trabalho** deverá responder livremente sendo o mais breve e directo possível.

Propriedades da imagem	Irrelevante	Pouco importante	Relevante	Importante	Vital	Não Sabe
<b>Tamanho:</b> Espaço que a imagem ocupa (por exemplo em disco )						
<b>Número de utilizadores:</b> Número de pessoas (clientes ou parceiros de trabalho) que utilizam este formato						
<b>Tempo de processamento:</b> Velocidade em que a imagem é aberta nos seus programas, velocidade de modificação e velocidade com que grava para ficheiro						
<b>Número de programas que suportam:</b> Número de programas que abrem, manipulam e gravam o formato de imagem (utilizados pelos clientes e parceiros)						
<b>Programas que suportam:</b> Programa específico que suporta o formato						
<b>Empresa que desenvolve:</b> Importância da empresa que lança, actualiza e eventualmente deixa de dar continuação a um formato de imagem						
<b>Transferência na Internet:</b> Facilidade e velocidade de transferência da imagem via Internet (para ou de clientes e parceiros)						
<b>Quantidade e qualidade de informação sobre o formato:</b> Importância sobre a quantidade de informação disponível para consulta pelo público geral sobre o formato e respectiva qualidade						
Formato não proprietário: Importância de ser um formato que não obriga a pagamentos para ser utilizado						

<b>Caracterização do trabalho:</b>	<b>Resposta</b>
Número de imagens criadas por semana:	
Número de imagens modificadas por semana:	
Número de imagens transferidas por semana: <b>recebidas</b>	
Número de imagens transferidas por semana: <b>enviadas</b>	
Espaço ocupado em disco pelas imagens que utiliza actualmente (tenha em conta o tamanho do seu arquivo de imagens)	

<b>Informação sobre preferências:</b>	<b>Resposta</b>
Preferência de formato(s):	
Preferência de Software(s):	

<b>Identificação do trabalho</b>	<b>Resposta</b>
Qual o seu trabalho:	

**Muito obrigado** pela sua cooperação, Filipe Quinaz.