

# Montagem de um SIG em software livre

Alexandre Hering Coelho

UFSC - Departamento de Engenharia Civil  
88040-900 Florianópolis, SC  
ahcoelho@web.de

**Resumo:** Se pode reconhecer hoje um aumento sensível na demanda por sistemas para auxiliar na análise de informações georreferenciadas. Simultaneamente ocorre um aumento no uso e no desenvolvimento do software livre. Este artigo procura, diante deste quadro, explicar o processo de montagem de um SIG em software livre, o mais completo possível, em uma estação de trabalho. Utilizando como componentes somente softwares de código aberto, é procurado reunir, instalar e testar um conjunto de softwares que sejam capazes de realizar operações básicas de um SIG. Operações de inserção e edição de dados raster e vetorial, processamento, consultas e visualização são buscadas para este fim. O grau de desenvolvimento do software, a consistência do sistema e a facilidade de uso são fatores que interferem decisivamente na escolha pelos componentes do SIG montado neste trabalho.

**Palavras-chave:** Sistemas de Informações Geográficas, software livre

**Abstract:** Today a sensible increment on the demand by systems for helping on the analysis of georeferenced information can be recognized. Simultaneously an increment in the use and in the development of open software occurs. This article tries, faced to this situation, to explain the assembly process of an open source GIS, as complete as possible, in a workstation. Using as components only open code software, it is aimed to bring together, to install and to test a group of softwares that is capable of carry out basic SIG operations. Operations of raster and vector data insertion and edition, processing, querying and viewing are taken as objective. The development niveau of the software, the consistency of the system and the facility of use are factors that affect decisively on the components selection for the SIG assembled in this work.

**Keywords:** Geographic Information Systems, open source

## 1. Introdução

É de grande importância, para o planejamento da ocupação da superfície terrestre, que informações georreferenciadas possam ser organizadas em Sistemas de Informações Geográficas (SIG). Nestes sistemas, dados alfanuméricos e geométricos são organizados e são oferecidas ferramentas para consultas, atualização, visualização e processamento, constituindo uma ferramenta poderosa para auxiliar nas tomadas de decisão. É grande e diversa a quantidade de informações que podem ser extraídas destes sistemas.

Existem vários softwares que auxiliam na realização das tarefas de SIG. Hoje em dia, os softwares mais consagrados mundialmente, em empresas, em universidades e em órgãos públicos, são patenteados e comercializados. Neste caso, os diversos pacotes de programas, em arquivos binários, resultantes da compilação do código fonte, são comercializados para sistemas operacionais específicos.

Porém, há também para este fim algumas alternativas interessantes em software livre, que são desenvolvidas por diversos grupos em todo o mundo. Os softwares livres também são expressivos na sua capacidade desenvolver tarefas de SIG. Segundo Kinberger & Puncher (2005), o uso de softwares livres na área de geoprocessamento está ganhando cada vez mais espaço em firmas e na administração pública, pois são capazes de substituir, na maioria dos casos, o uso de softwares comerciais.

Por *software* livre são compreendidos, neste trabalho, os *softwares* cujos códigos fonte são distribuídos gratuitamente, normalmente licenciados pela *GNU General Public License* ou pela *LGPL (GNU Lesser General Public License)*.

O uso de *softwares* livres oferece a vantagem de não haver custos financeiros por compra de *software*. Isto pode ser vantajoso em certas situações, por exemplo quando um grande número de cópias deve ser instalado, em várias estações de trabalho. Além disso, o *software* livre pode oferecer a vantagem de suprir atualizações de *software* mais facilmente, perante a rápida evolução do *hardware*. Porém, no que se refere à mão de obra para instalação, treinamento, e manutenção, é atualmente mais difícil encontrar profissionais qualificados em *softwares* livres do que em *softwares* patenteados.

No desenvolvimento de *softwares* específicos para SIG, sejam livres ou proprietários, podem ser percebidas duas correntes claras, também relacionadas à licença para uso. Por um lado, se coloca o desenvolvimento de sistemas completos e complexos, que são capazes de realizar quase todas as funções envolvidas, oferecendo interfaces gráficas amigáveis para os usuários. Esse é o caso dos *softwares* proprietários. No *software* livre, o desenvolvimento ocorre de forma diferente. A iniciativa é normalmente financiada por fundos de pesquisa e desenvolvimento, tanto em empresas como em universidades e órgãos públicos, em todo o mundo. Além disso, há também os entusiastas programadores, que contribuem também de maneira decisiva. O desenvolvimento fica assim mais pulverizado, sendo criados *softwares* normalmente de menor porte, voltados para necessidades mais específicas. Câmara & Onsrud (2004) chegaram, em sua análise, a uma conclusão semelhante.

A característica do desenvolvimento pulverizado não implica, porém, na incompatibilidade entre as diversas ferramentas criadas. Uma prova disso é a existência do OpenGIS (*OpenGIS Consortium, Inc*, 1999), que é um padrão para a linguagem SQL que define o suporte para o armazenamento, consulta e atualização de dados georreferenciados via *Open Data Base Connectivity* (ODBC). Uchoa & Ferreira (2004), somam a este fator a revolução do *software* livre, apoiada pela *Free Software Foundation*, que promove o seu uso e seu o desenvolvimento.

Estes, entre outros fatos e discussões relacionados com o tema, tornam interessante a identificação de um conjunto de *softwares* livres que, trabalhando de forma integrada, seja capaz de armazenar dados georreferenciados e oferecer ferramentas para gerenciamento, consultas, processamento e visualização de informações alfanuméricas e geométricas, constituindo um SIG. Com este trabalho, é buscado satisfazer esta tarefa, onde possíveis soluções para estações de trabalho (*desktops*) são analisadas.

Partindo do princípio que a falta de conhecimento em *software* livre, seja pela falta de documentação ou pela maior dificuldade associada, é um fator que dificulta a decisão por esta alternativa, os *softwares* escolhidos neste trabalho deverão ser de uso o mais amigável e intuitivo possível. Isto implica em estarem disponíveis interfaces gráficas para a realização do maior número de tarefas possível, evitando o uso de terminais de linha de comando. Além disso, o grau de desenvolvimento do *software* e a consistência do sistema montado também são fatores decisivos na escolha. Ao final do trabalho, deve estar caracterizado um SIG completo, constituído exclusivamente de *softwares* livres.

Na seção 2 a seguir são apresentados os *softwares* escolhidos para compor o SIG, suas descrições e as características principais de cada um. Na seção 3 é abordada a instalação dos *softwares* escolhidos, montando o SIG e preparando-o para uso. Na seção 4 são demonstradas e ilustradas algumas das funcionalidades principais do SIG utilizando os *softwares* instalados. Finalmente, nas seção 5, são discutidas e avaliadas as eventuais dificuldades encontradas, as vantagens identificadas no uso do SIG em *software* livre e apresentadas outras considerações.

## 2. Busca por *softwares*

O processo de busca por *softwares* pode ser feito com o auxílio de uma ferramenta de busca na *internet*. Os grupos desenvolvedores de *software* livre mantém normalmente páginas na *internet* onde é possível obter o código fonte dos *softwares*, pacotes pré-compilados para sistemas operacionais específicos, documentação, entre outros recursos. É grande a quantidade de *softwares* que pode ser encontrada desta forma. O trabalho de eleição de um *software*, de acordo com os critérios apresentados, implica na sua instalação e na realização de testes com alguns dados.

Podem ser estabelecidos três grupos principais de *softwares*, que trabalhando de forma conjunta são capazes de constituir um SIG completo. São eles os *softwares* específicos para operações de SIG, os *softwares* para bancos de dados e os *softwares* de apoio.

Primeiramente, foram buscados *softwares* específicos para SIG, que englobam funções de leitura e gravação de dados, ferramentas para consultas, visualização e processamento. Foram escolhidos, dentre vários outros *softwares* analisados, o Quantum GIS (versão 0.8.0 *Preview 1*, obtida em <http://download.qgis.org>) e o GRASS (versão 6.0.1, obtida em <http://grass.itc.it/download/index.php>).

O Quantum GIS é um SIG com interface gráfica amigável para o usuário, com características que competem principalmente com o *software* comercial ArcView, da firma ESRI. É capaz de carregar, utilizando bibliotecas de apoio, uma grande variedade de dados, em formato raster e vetorial, e tem grande capacidade de edição de propriedades visuais de *layers*. Possibilita a edição de informações alfanuméricas e geométricas de *layers* vetoriais em arquivos e diretamente no banco de dados. Oferece ainda uma ferramenta para a confecção de mapas impressos (*layouts*).

O GRASS (*Geographic Resources Analysis Support System*) é um SIG mais completo, englobando mais de 350 funções para análise geoespacial, modelagem ambiental, mapas temáticos, integração de banco de dados, processamento de imagens e visualização. Possui também interface gráfica bem desenvolvida para utilização, mas ela não é tão intuitiva como no Quantum GIS. O GRASS é normalmente utilizado para executar operações de SIG mais complexas. De acordo com Kinberger & Pucher (2005), as possibilidades oferecidas pelo GRASS são hoje até mesmo dificilmente alcançadas com o uso de *softwares* proprietários, onde o suporte de GIS para dados raster e vetorial são combinadas com técnicas de visualização e processamento de imagens. Possui manual completo de utilização, elaborado por GDF Hannover (2005). Segundo Smotritsky (2004) o GRASS é o maior, mais poderoso e mais confiável *software* livre para SIG que existe hoje.

De acordo com Bähr (1999), os SIGs integram, além de ferramentas para consulta, atualização, visualização e processamento de dados georreferenciados, os bancos de dados geográficos como um outro componente básico. Segundo o autor, pelo uso de bancos de dados geográficos, é possível interligar dados semânticos (alfanuméricos) a elementos geométricos representados em um sistema de coordenadas homogêneo. Os resultados de consultas em SIGs podem ser diretamente visualizados em forma de mapas e também é possível a geração de relatórios.

A alternativa mais atraente e mais adotada em banco de dados livre (por exemplo em Uchoa & Ferreira, 2004; Kinberger & Pucher, 2005; GDF Hannover, 2005) é o PostgreSQL (versão 8.1.3, obtida em [www.postgresql.org](http://www.postgresql.org)), pela sua robustez, seu desempenho e por sua compatibilidade com os *softwares* livres para SIG. Outra grande vantagem deste bando de dados relacional é a existência do módulo espacial PostGIS (versão 1.4.4, obtida em <http://postgis.refractive.net>), que possibilita o armazenamento de informações geométricas e a realização de consultas envolvendo análises de geometria.

Como *softwares* de apoio foram integrados uma biblioteca para tratamento de projeções cartográficas (PROJ, versão 4.4.9 obtida em <http://proj.maptools.org>), bibliotecas para leitura e gravação de dados em diversos formatos raster e vetorial (GDAL/OGR, versão 1.3.2 obtida em <http://www.gdal.org>), um *software* para ligação dos SIGs com o banco de dados (GEOS, versão 2.1.4 obtida em <http://geos.refractive.net>) e um *software* para processamento numérico (Octave, versão 2.1.57-2 obtida em <http://www.octave.org>).

O sistema operacional Linux foi utilizado como plataforma para compilação, instalação e utilização dos *softwares*.

### 3. Montagem do SIG

Na montagem do SIG, descrita neste trabalho, são instalados os *softwares* e utilizados alguns dados, para a realização dos testes.

#### 3.1 Compilação e instalação dos *softwares*

Para todos os *softwares* empregados são disponibilizados pelos desenvolvedores pacotes pré-compilados, que podem ser diretamente instalados no sistema operacional. Porém, se forem instalados desta forma, não fica estabelecida a interligação entre eles. Para que todos os componentes possam se comunicar entre si, utilizando todo o potencial das ferramentas e tornando o SIG capaz de realizar operações mais

complexas, é necessária a instalação de cada componente a partir do seu código fonte.

O procedimento de instalação via código fonte implica no uso de terminais de linha de comando, onde é feita a configuração dos arquivos que gerenciam a compilação, seguida da compilação e da instalação. Os *softwares* devem ser compilados e instalados em uma sequência específica, de forma que *softwares* posteriormente instalados sejam capaz de incorporar as funcionalidades dos *softwares* já instalados. A sequência de instalação dos *softwares* é obtida durante a configuração para compilação dos componentes, onde as dependências são acusadas. Além dos *softwares* que compõem o SIG, surgem também outras dependências, de *softwares* e bibliotecas, que devem ser, da mesma forma, instalados.

### 3.2 Obtenção de alguns dados

Alguns dados, para serem utilizados nos testes, foram obtidos também na *internet*.

Como dados vetoriais foram utilizados um arquivo *shape* com a malha municipal do Brasil, sedes dos municípios, população, área, nome e código, e um arquivo *shape* com dados sobre a hidrografia brasileira. Os arquivos foram obtidos na página da empresa Gismaps (<http://www.gismaps.com.br/divpol/divpol.htm>).

Como dados em formato raster foi utilizado um modelo digital de elevação, obtido na página da U.S. Geological Survey (GTOPO30, <http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html>), em formato \* .DEM. As informações sobre projeção cartográfica são também disponibilizadas, em arquivos \* .PRJ.

### 3.3 Criando a *Location* no GRASS

Os dados no GRASS são armazenados em um diretório denominado DATABASE. Este diretório deve ser criado manualmente, antes de começar o trabalho com o GRASS. Dentro do diretório DATABASE, os projetos são organizados por áreas de projeto armazenadas em sub-diretórios denominados *Locations*.

Uma *Location* é definida pelo seu sistema de coordenadas, projeção do mapa e limites geográficos. Os sub-diretórios e arquivos que definem uma *Location* são criados automaticamente quando o GRASS é iniciado pela primeira vez com uma nova *Location*.

O sistema de coordenadas e a projeção do mapa podem ser obtidas com o auxílio do Quantum GIS, carregando o arquivo *shape* ou o DEM e consultando as suas propriedades. Os limites geográficos também podem ser obtidos com o Quantum GIS, onde as coordenadas norte, sul, leste e oeste são indicadas passando o *mouse* sobre o mapa, na tela do computador. No caso, os arquivos *shape* e DEM estão georreferenciados dentro da mesma projeção cartográfica (coordenadas geográficas) e se baseiam no mesmo datum (WGS84).

Com estas informações, pode ser criada a *Location*. O procedimento é auxiliado por uma interface em ASCII, em um terminal de linha de comando. Durante a utilização do GRASS fica também disponível um terminal de linha de comando, onde as várias funções podem ser acessadas, ao serem digitados os seus nomes. As funções podem também ser acessadas através da interface gráfica principal do GRASS.

Foi criada uma *Location* para a área do estado de Santa Catarina.

## 4. Algumas demonstrações de funcionalidades

Nesta seção são demonstradas algumas funcionalidades, possíveis de serem desempenhadas com o SIG montado. Não há como demonstrar neste trabalho todas as ferramentas disponibilizadas pelos *softwares*, porém, é buscado ilustrar algumas das funções principais e suas interfaces gráficas, em funcionamento.

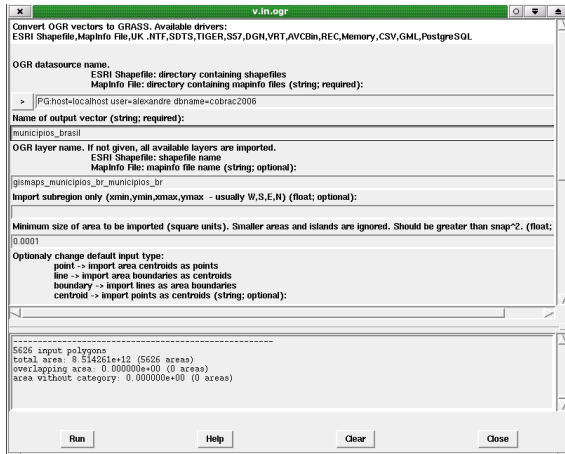
### 4.1 Inserção de dados

No Quantum GIS, o carregamento dos arquivos raster e vetorial pode ser feito de maneira simples e direta, utilizando os ícones na barra de ferramentas.

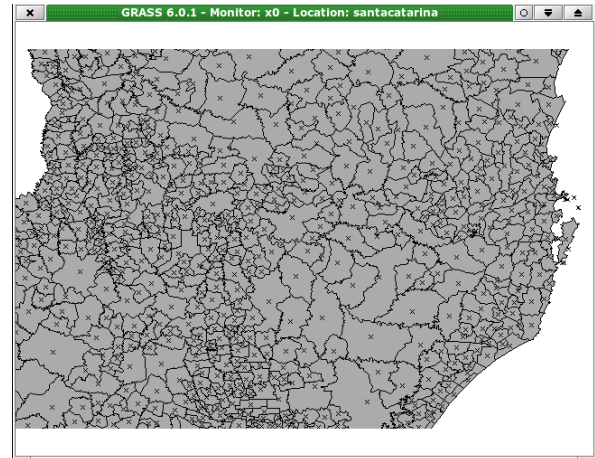
O carregamento dos arquivos no GRASS é feito através das funções `v.in.ogr` para dados vetoriais e da função `r.in.gdal` para dados raster. Quando estas funções são utilizadas, somente a informação que

está contida dentro da área da *Location* é carregada a partir dos arquivos. A Figura 1 ilustra as interfaces gráficas das duas funções e a visualização dos dados raster e vetoriais carregados.

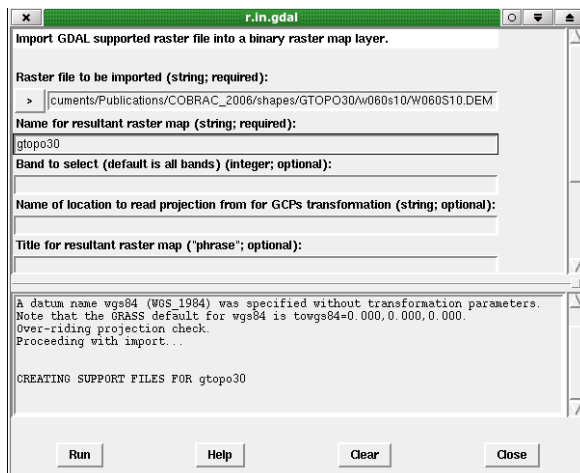
A inserção de dados vetoriais e de seus atributos alfanuméricos no banco de dados PostgreSQL é feita utilizando a função `shp2pgsql` do PostGIS, o que pode ser feito somente a partir de terminal de linha de comando. Apesar de não haver interface gráfica para esta operação, a possibilidade de usar a função embutida em um *script* facilita a inserção de vários arquivos de maneira sequencial, com uma só operação.



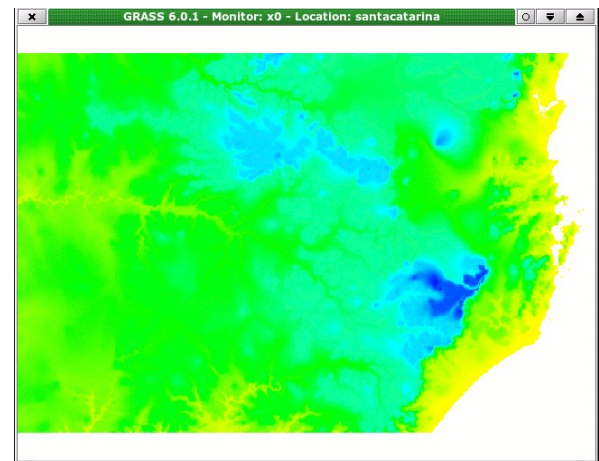
(a) Interface gráfica da função `v.in.ogr`.



(b) Visualização dos dados dos municípios brasileiros (vetorial) carregados no GRASS.



(c) Interface gráfica da função `r.in.gdal`.



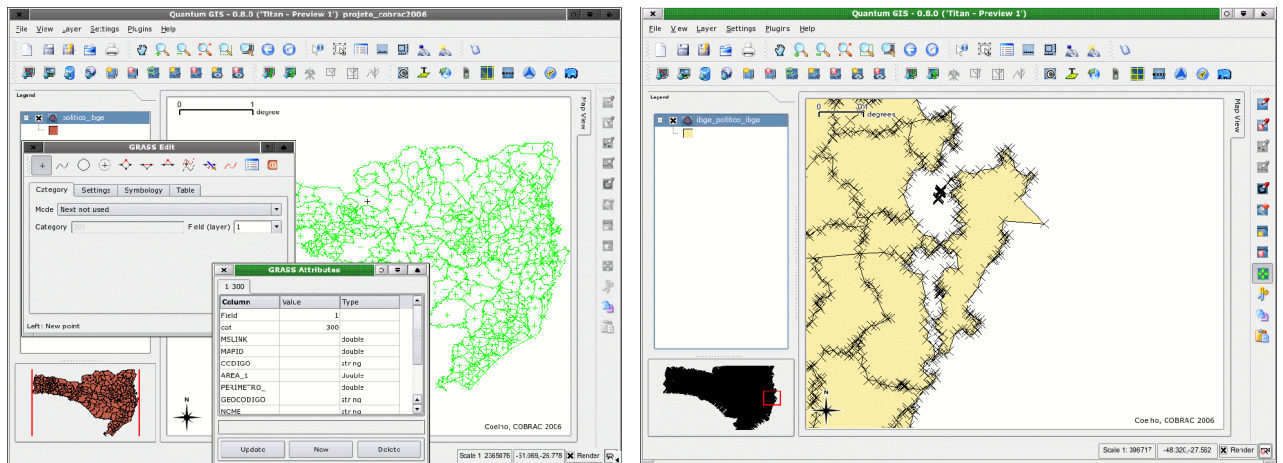
(d) Visualização dos dados de elevação (raster) carregados no GRASS.

**Figura 1** : Inserção de dados raster e vetorial no GRASS.

## 4.2 Edição de dados

Como exemplo para a edição de dados foi procurado editar visualmente dados vetoriais. Para esta tarefa se mostra mais adequado o Quantum GIS, que oferece ferramentas de edição tanto para *layers* vetoriais do GRASS quanto do PostGIS.

Na Figura 2 é mostrada a interface gráfica do Quantum GIS durante a realização destas tarefas. Na Figura 2(a) é editado um *layer* do GRASS com as informações geométricas dos limites políticos dos municípios de Santa Catarina, carregado com o auxílio de um *plugin* no Quantum GIS. Na Figura 2(b) é ilustrada a edição do mesmo *layer*, que foi armazenado no PostgreSQL. Há, em ambos os casos, ferramentas para a inserção e edição de pontos e vértices de polígonos. No caso de inserção de novas entidades geométricas, o Quantum GIS oferece também interface gráfica para a inserção de atributos alfanuméricos respectivos.



(a) Layer do GRASS.

(b) Layer do PostGIS.

**Figura 2 :** Edição de dados vetoriais em *layers* do GRASS e do PostGIS no Quantum GIS.

### 4.3 Consultas

O Quantum GIS possibilita, durante o carregamento de *layers* do PostGIS, selecionar os dados a serem carregados através de uma cláusula de consulta. Isto é feito especificando também uma cláusula *where* para o comando *select* do banco de dados. Usando esta possibilidade, podem ser carregados no Quantum GIS, por exemplo, somente as informações geométricas e alfanuméricas dos municípios que fazem parte de Santa Catarina. Isso é conseguido através da cláusula “*where nomeuf='SANTA CATARINA'*”. O nome da coluna usada para a filtragem pode ser obtida consultando a tabela de atributos original do *layer*, no Quantum GIS.

É possível serem criadas consultas onde estão contidas funções do PostGIS, como também cláusulas envolvendo o conteúdo de várias tabelas, o que permite obter resultados capazes de responder a questões bem específicas. Existindo também relacionamentos entre as tabelas, definidas no banco de dados, elas também influenciam nos resultados das consultas. Porém, a construção do comando *select* pode ficar, assim, bastante complexa.

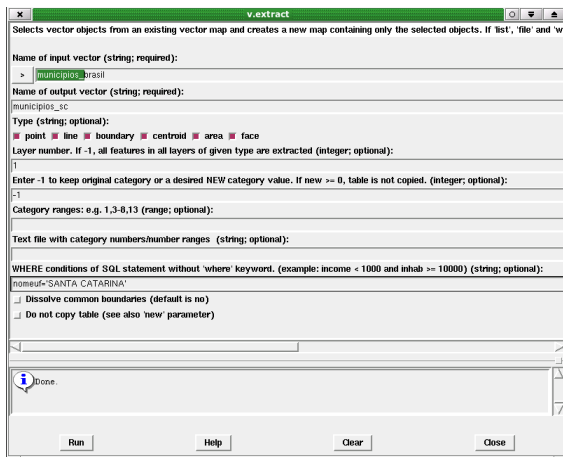
O Quantum GIS oferece uma ferramenta gráfica que auxilia na construção da cláusula *where* para o comando *select*. Porém, esta ferramenta se limita a buscar informações contidas dentro de somente um tabela. Para realizar consultas mais complexas, envolvendo várias tabelas, é necessário construir comandos *select* completos externamente, como o auxílio de algum outro recurso, como por exemplo, usando o *software* Knoda (<http://www.knoda.org>). Este *software*, porém, tem também as suas limitações. Para a realização de uma consulta com nível de complexidade maior, envolvendo também funções do PostGIS, não há ainda como evitar a construção manual do comando *select*, o que exige o domínio da linguagem SQL.

Outra limitação do Quantum GIS é a impossibilidade de gravação do resultado da consulta em um novo *layer*. Isto dificulta a extração de informações dos dados, uma vez que o mapa temático ou relatório resultante deve ser integralmente formado através de uma única consulta ao banco de dados. Análises que, por exemplo, envolvam conversão de dados raster para vetorial, e vice-versa, para se chegar aos resultados esperados, fica desta forma impossibilitada.

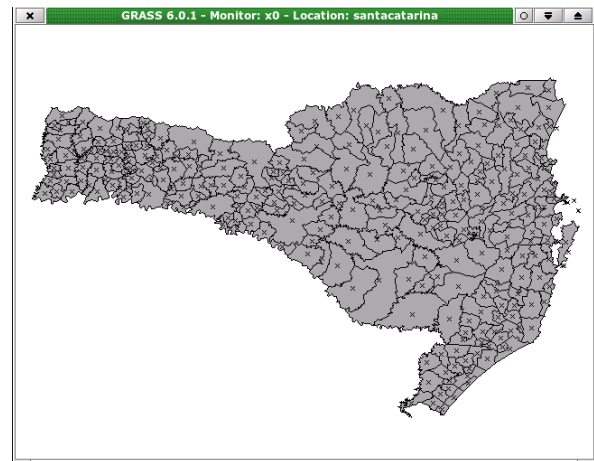
O GRASS, em contrapartida, apresenta soluções para estes dois problemas. Com o auxílio da função *v.extract*, cuja interface gráfica é apresentada na Figura 3(a), é criado um novo *layer*, contendo somente as informações (geométricas e alfanuméricas) resultantes da consulta, onde a cláusula *where* pode ser especificada. Não há no GRASS nenhuma ferramenta para auxiliar visualmente na montagem desta cláusula. Por um lado isso dificulta o uso do *software* para usuários mais inexperientes, mas por outro lado, força o conhecimento da linguagem SQL, o que é muito valioso quando se trabalha com SIGs.

No caso do exemplo mostrado, o novo *layer* contém somente os dados sobre os municípios de Santa Catarina, como mostra a Figura 3(b).

A função `db.execute` oferece uma interface gráfica onde pode ser inserido um comando `select` completo. A criação de um novo *layer*, contendo o resultado deve, porém, estar também inserida no comando SQL. Há no GRASS várias outras funções para consulta e manipulação de dados em banco de dados, como por exemplo `db.connect`, `db.copy`, `db.describe`, `db.select` e `v.select`, que associadas ao conhecimento da linguagem SQL constituem uma ferramenta completa para a análise, cruzamento e extração de informações e geração de mapas temáticos.



(a) Interface gráfica da função `v.extract`



(b) Conteúdo do novo *layer*

**Figura 3** : Aplicação da função `v.extract` sobre os dados vetoriais, criando um novo *layer* vetorial contendo os dados que obedecem à cláusula `where nomeuf = 'SANTA CATARINA'`.

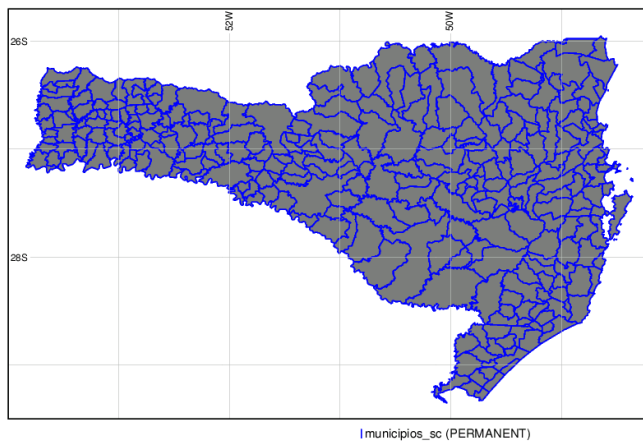
#### 4.4 Visualização

Tanto o Quantum GIS como o GRASS possuem funcionalidades para visualização, porém eles têm características bem diferentes. As ferramentas presentes nos dois *softwares* são quase que complementares. O Quantum GIS se preocupa mais com a parte de visualização 2D em tela, oferecendo uma maneira fácil de construção elaborada do mapa, com escala, norte, legenda, e paleta de cores, entre outros. Já o GRASS não dispõe de facilidades para a confecção mais elaborada de um mapa em tela, com todos os recursos que o Quantum GIS oferece, porém possui uma função para visualização 3D chamada `Nviz`. A Figura 4(b) ilustra a visualização dos dados raster, como modelo digital de elevação, e os dados vetoriais de limites de municípios, sobrepostos em uma representação tridimensional. Este recurso é interessante, porém seria mais completo se fosse possível também inserir facilmente elementos como legendas e representação do sistema de coordenadas de referência.

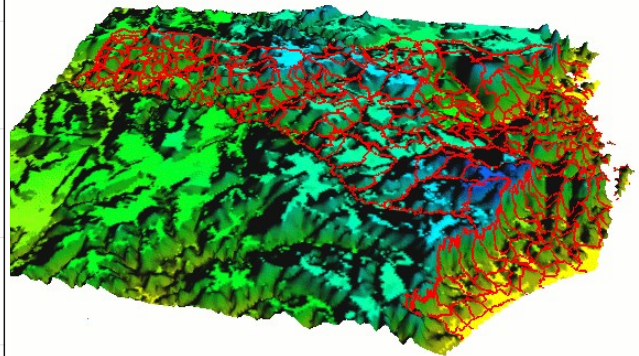
Ambos os *softwares* oferecem ferramentas para a edição de *layout* de mapas para impressão, porém, nenhuma delas alcança a facilidade de uso e a qualidade que pode ser obtida usando *softwares* proprietários. No Quantum GIS, o processo de confecção de *layout* é muito facilitado, estando disponível uma interface gráfica para edição visual. Porém, os recursos são ainda muito limitados, sendo possível criar um produto final também limitado. Com o GRASS, *layouts* mais completos podem ser criados. O problema deste *software* reside na dificuldade da edição, que pode somente ser feita através da criação de um arquivo com várias instruções, que é processado pelo comando `ps.map`, que gera o mapa em formato PostScript. O grau de desenvolvimento das ferramentas no GRASS poderia ser maior, se abrangesse também a facilidade de uso. A Figura 4(a) ilustra um *layout* de mapa para impressão criado com o GRASS, onde somente as variáveis `vareas`, `vlegend` e `grid`, foram empregadas no arquivo de instruções.

Com o GRASS, é possível ainda usar a função `d.out.png` para gerar uma imagem, que pode ser carregada em um outro *software* para edição do *layout* de um mapa. O software XFig (<http://www.xfig.org>) é indicado para este fim, sendo recomendado no próprio manual do GRASS (GDF Hannover, 2005). De

qualquer forma, o processo de confecção de mapas no GRASS é bastante complexo. Porém, uma vez que estejam prontos os arquivos de configuração do `ps.map`, ou o `layout` no XFig, o processo de produção se torna sistemático.



(a) Figura gerada com a função `ps.map`.



(b) Representação tridimensional da divisão política (vetorial, em vermelho) sobre o relevo de Santa Catarina (raster) usando o `Nviz`.

**Figura 4** : Possibilidade de confecção de mapas para impressão e representação 3D de dados usando ferramentas do GRASS.

#### 4.5 Processamento

No âmbito do processamento de dados, a alternativa é o uso do GRASS. E são várias as possibilidades para processamento oferecidas pelo *software*, que envolvem filtragem e classificação de dados raster, aplicação de topologia a dados vetoriais, entre outras. Para exemplificar um caso de processamento, é obtido, a partir dos dados raster de modelo de elevação, um *layer* vetorial contendo dados geométricos de possíveis leitos de rios.

O procedimento envolve o uso de algumas funções do GRASS, utilizadas em uma certa sequência. Inicialmente, com o uso da função `r.watershed` é obtido, em formato raster, a informação dos possíveis leitos de rios, a partir dos dados de elevação. O resultado deste processamento está ilustrado na Figura 5(a). Então, é feita a conversão desta informação para o formato vetorial. Isto pode ser feito com o auxílio das funções `r.thin` e `r.to.vect`, em sequência. Porém, utilizando diretamente a função `r.thin` sobre o *layer* raster criado pela função `r.watershed`, o número de linhas criadas no *layer* vetorial pela função `r.to.vect` é demasiado grande. É necessário primeiramente, então, realizar uma filtragem nos *layer* raster.

Com o auxílio da função `r.out.ascii` é possível exportar o *layer* raster criado pela função `r.watershed` para um arquivo ASCII, contendo as coordenadas norte, sul, leste e oeste da área, o número de linhas e colunas que formam o *layer* raster e os valores de elevação. Este arquivo pode ser então processado usando o *software* Octave. Com o objetivo de reduzir o número de valores que um pixel pode assumir para somente dois, realizando uma binarização da imagem, é buscado atribuir o valor 1 para os pixels que se encontram nos leitos de rios e 0 para os demais. Para realizar este processamento, o seguinte arquivo de instruções foi dado ao *software* Octave:

```
%lendo arquivo com dados do watershed
load -ascii -force data_in.dat data_in

%extraíndo o número de linhas e colunas da matriz
nl=rows(data_in);
nc=columns(data_in);

%atribuindo o valor 1 para valores maiores que 40 e 0 para os
demais
for i=1:nl;
    for j=1:nc;
```



```

if data_in(i,j) >= 40;
    data_out(i,j)=1;
else;
    data_out(i,j)=0;
end;
end;
end;

%salvando o resultado no arquivo data_out.out
save -ascii data_out.out data_out

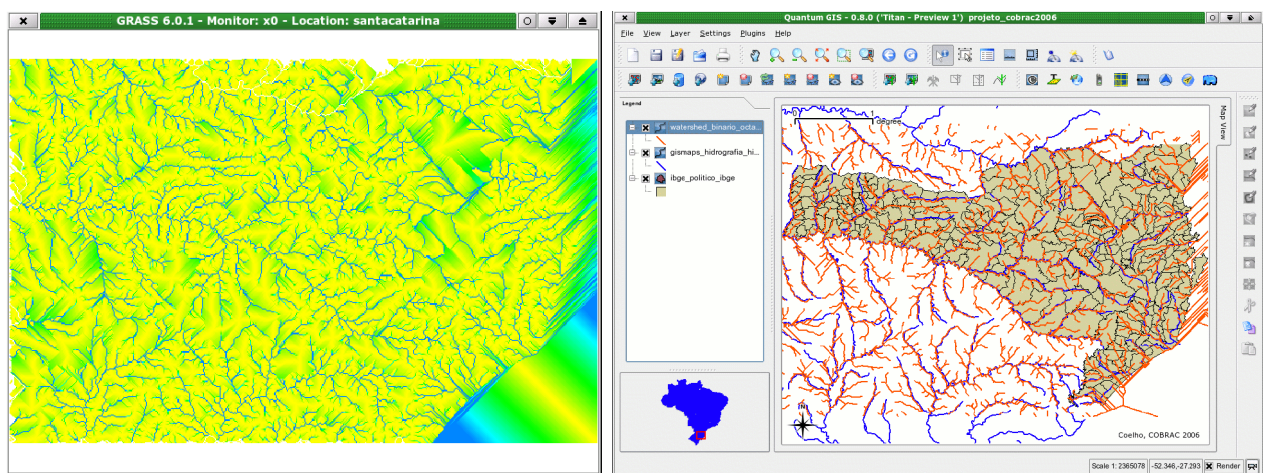
```

No caso, todos os pixels que tiverem valores maiores ou iguais a 40 recebem o valor 1 e aos demais é atribuído o valor 0. O valor limite de 40 foi obtido em uma análise dos valores máximos e mínimos existentes no *layer* raster, com o auxílio da função `r.univar` do GRASS.

O arquivo resultante pode ser então retornado para o GRASS com o uso da função `r.in.ascii`, pela qual um *layer* raster pode ser lido a partir de um arquivo ASCII contendo uma matriz com informações de valores de pixels.

Os cabeçalhos dos arquivos gerados pela função `r.out.ascii` e lidos pela função `r.in.ascii` possuem conteúdos diferentes dos necessários para o processamento com o Octave. Por isso é necessária uma edição manual, usando algum *software* para edição de arquivos ASCII.

O *layer* importado pode ser agora processado pelas funções `r.thin` e `r.to.vect`. O resultado final é mostrado na Figura 5(b), onde o *layer* vetorial resultante do processamento é sobreposto ao *layer* vetorial contendo a malha hidrográfica na área, obtido na página da firma GisMaps.



(a) Resultado do processamento com a função `r.watershed` (raster).

(b) Visualização do *layer* vetorial resultante no Quantum GIS. Em azul os dados de hidrografia da empresa GisMaps e em laranja o resultado final do processamento.

**Figura 5** : Ilustração dos resultados do processamento para a derivação de possíveis leitos de rios a partir de dados de elevação.

## 5. Avaliação e conclusões

A existência do *software* livre oferece uma gama maior de ferramentas que podem ser utilizadas para solucionar problemas de SIG. O atual grau de desenvolvimento dos *softwares* livres, que compõem o SIG montado neste trabalho, já oferece a possibilidade de realizar todas as tarefas que são atribuídas a este tipo de sistema.

Não há um *software* livre que seja capaz de realizar sozinho todas as tarefas englobadas em um SIG. É

necessário definir um grupo de *softwares* que trabalhem de forma conjunta e complementar. Este trabalho mostrou que isto é possível, uma vez que as faltas de um componente podem ser supridas por funcionalidades de outro, e que os componentes são capazes de se comunicar entre si, com o uso de *softwares* e bibliotecas de apoio.

O número de *softwares* livres que se dispõem a realizar tarefas de SIG é grande. Há muitas possibilidades para a escolha, sendo que cada uma possui vantagens específicas. Também há vários graus diferentes de desenvolvimento entre elas.

Levando em consideração os requisitos estabelecidos para a escolha dos *softwares* para constituir um SIG completo, que são a disponibilidade dos códigos fonte, o grau de desenvolvimento, a estabilidade do sistema e a facilidade de uso, os componentes escolhidos foram o Quantum GIS, o GRASS, o PostgreSQL/PostGIS, PROJ, GDAL/OGR, GEOS e Octave, que foram compilados e instalados no sistema operacional Linux.

A maioria das operações desenvolvidas com o SIG montado pode ser realizada com o auxílio de interfaces gráficas. Porém, algumas operações importantes devem ser, ainda, realizadas via terminal de linha de comando.

Deve haver, entre os *softwares* livres, uma maior preocupação com a possibilidade de edição de *layout* de mapas para impressão. Apesar de haver hoje um grande interesse em serem utilizadas e desenvolvidas soluções para *internet*, os mapas impressos nunca perderão a sua utilidade.

Apesar do tema específico deste trabalho tratar da montagem de um SIG completo em uma estação de trabalho, os mesmos *softwares* podem ser utilizados para a formação de um grupo de trabalho. A peça chave neste aspecto é o uso do banco de dados, que pode armazenar as informações em um servidor de dados central e disponibilizá-las para várias estações de trabalho.

Não há como definir o uso do *software* livre ou proprietário como sendo o melhor para qualquer caso. A escolha deve ser muito bem estudada, para cada caso particular. Porém, como foi demonstrado neste trabalho, se for optado pelo uso do *software* livre, uma grande quantidade de possibilidades para a organização de dados, consultas, edição, visualização e processamento é oferecida.

## Referências bibliográficas

- Bähr, H.-P.:** *GIS for Environmental Monitoring*. Cap. GIS Introduction. Schweizerbart, Stuttgart, Alemanha, 1999, p. 1-9.
- Câmara, G.; Onsrud, H.:** *Open source GIS software: myths and realities*. In: Julie M. Esanu and Paul F. Uhler, Eds, *Open Access and the Public Domain in Digital Data and Information for Science: Proceedings of an International Symposium*. Washington, The National Academies Press, 2004. (ISBN 0-309-09145-4).
- GDF Hannover:** *An introduction to the practical use of the Free Geographical Information System GRASS 6.0*. Version 1.2, 2005. Published under GNU FDL, <http://www.gdf-hannover.de/literature>
- Kinberger, M.; Pucher, A.:** *Open Source GIS als Alternative im Desktop-Bereich - Evaluation freier Software im Bereich Geoinformation*. In: CORP 2005. Wien, Österreich, Februar 2005.
- OpenGIS Consortium, Inc:** *OpenGIS Simple Features Specification for SQL*. Revision 1.1, 1999. Disponível em: <http://www.opengeospatial.org/docs/99-049.pdf>
- Smotritsky, Y.:** *Running GIS on open source*. In: Connect - Information Technology at NYU. New York University, 2004. Disponível em: [http://www.nyu.edu/its/pubs/connect/fall04/smotritsky\\_GIS.html](http://www.nyu.edu/its/pubs/connect/fall04/smotritsky_GIS.html)
- Uchoa, H. N.; Ferreira, P. R.:** *Geoprocessamento em Software Livre*. Versão 1.0 de 26 de outubro de 2004. Disponível em: [http://www.igc.usp.br/pessoais/guano/downloads/geoprocessamento\\_software\\_livre\\_Uchoa-Roberto-v1.0.pdf](http://www.igc.usp.br/pessoais/guano/downloads/geoprocessamento_software_livre_Uchoa-Roberto-v1.0.pdf).