

## DESENVOLVIMENTO DO PROGRAMA WINGIS PARA ANÁLISE DE INFORMAÇÕES GEOGRÁFICAS

**Dhiancarlo Macedo Pinheiro & Nilton Correia da Silva**

Centro Universitário de Anápolis (Unievangélica)  
Avenida Universitária km 3,5, Cidade Universitária, Anápolis, GO, Brasil.  
nilton@unievangelica.edu.br

Recebido 01 de junho de 2009; revisado 03 de agosto; aceito 13 de agosto.

**RESUMO** – O presente artigo tem como objetivo utilizar um *framework* para desenvolvimento de aplicações (QT) visando à elaboração de interfaces para visualização integrada de dados espaciais vetoriais. Para tanto, esta pesquisa baseia-se nos formalismos matemáticos referentes a coordenadas plano retangulares e implementação de visualizações bidimensionais de dados vetoriais utilizando funções do *MapServer*. Como produto final, foi desenvolvido o protótipo WinGis que consiste em um ambiente amigável para leitura, visualização e integração de dados espaciais vetoriais. Este protótipo permite realizar uma análise conjunta de diferentes informações espaciais (pontos, linhas e polígonos) de um dado recorte espacial.

**Palavras-Chave** – QT, shapefiles, MapServer, Interface, SIG.

**ABSTRACT** – This article aims to use a framework for application development in order to build up interfaces for integrated visualization of spatial data vector. Thus, this research is based on mathematical formalisms related to rectangular coordinates and implementation of two-dimensional visualizations of vector data using MapServer functions. As a final product, we developed the prototype WinGis which consists of a friendly environment for reading, viewing and integration of spatial data vector. This prototype provides a conjoint analysis of different spatial information (points, lines and polygons) of a given spatial area.

**Keywords** - QT, shapefiles, MapServer, Interface, GIS.

## INTRODUÇÃO

O Sistema de Informações Geográficas (SIG) é uma ferramenta que provê mecanismos para modelagem dos complexos processos do mundo real e do ambiente humano em certos níveis de relação, simplificação, generalização e abstração (Paredes, 1994; Teixeira et al., 1992). Nesses sistemas os dados (ou entidades) estão arrançados de modo a interagir e subsidiar análises de problemas espaciais. Desta forma, o emprego do SIG torna-se uma ferramenta decisiva na gestão do território por proporcionar um conhecimento amplo, atualizado e bem controlado de uma área de estudo.

Conforme Rosa (1992) o SIG possui os seguintes objetivos: (a) integrar informações que representem os vários aspectos do estudo de uma determinada região em uma única base de dados; (b) permitir a entrada de dados de diversas formas (mesas digitalizadoras, “scanners”, teclado); (c) combinar dados de diversas fontes (dados cartográficos, de censo, de cadastro urbano ou rural, imagens de satélite, modelos digitais de elevação), produzindo novos tipos de informação; e (d) gerar relatórios e documentos gráficos dos mais variados tipos.

Dentro da tecnologia dos SIGs possibilita-se criar um banco de dados codificado espacialmente onde é permitido realizar operações algébricas entre mapas. A “álgebra de mapas” é um conjunto de procedimentos de análise espacial que produz novos dados, a partir de funções de manipulação aplicadas a um ou mais mapas.

O aumento da demanda na utilização dessa tecnologia para análise de dados geoespaciais e a importância de se contar com uma boa ferramenta de desenvolvimento para este fim, justificam a necessidade de se pesquisar novas ferramentas para o seu desenvolvimento.

O presente trabalho possui como objetivo desenvolver um programa para atender as funcionalidades de visualização e de manipulação básicas de um SIG usando tecnologias livres. O fato de desenvolver um SIG baseado em programa livre e de código aberto faz com que diminua os custos e que este permita o constante desenvolvimento de módulos futuros por outros pesquisadores. A pesquisa realizada adota a ferramenta QT e sua aplicação em sistemas geográficos juntamente com dois componentes auxiliares, a biblioteca ShapeLib (escrita em linguagem C) e um servidor web para construção de mapas chamado MapServer. O QT foi escolhido por ser um ambiente de desenvolvimento em linguagem C/C++ gratuito, multiplataforma, que permite uma direta integração com a biblioteca MapServer e também possui uma interface de desenvolvimento com componentes visuais, o que permite uma maior velocidade na fase de desenvolvimento. Desta forma, utilizando a linguagem c++ na plataforma QT, foi desenvolvido o WinGis, um sistema para visualização integrada (em camadas) de dados vetoriais (pontos, linhas e polígonos) geoespacializados.

## **MATERIAIS E MÉTODOS**

### **ARQUIVOS VETORIAIS (SHAPEFILES)**

Os arquivos shapefiles armazenam tipos dados geométricos primitivos de linhas, pontos e polígonos. O formato Shapefile está publicamente documentado pela

Esri, que criou uma padronização chamada de Manifesto Shapefile. Este manifesto define padrões que associam os dados geométricos com a biblioteca XBASE, esta permite a manipulação de arquivos.dbf.

Um Shapefile é composto de três arquivos:

- shp — é a informação geométrica dos dados espaciais do shape;
- dbf — atributo colunar de cada figura geométrica do shape (dBase IV)
- shx — contém o índice posicional dos shapex e cria o vínculo entre o arquivo shp e o arquivo dbf ;

## SHAPELIB

A Shapelib é uma biblioteca escrita em C, gratuita e multiplataforma. Permite ler, escrever e modificar os arquivos shapefiles. Está presente em várias aplicações GIS, entre elas o QuantumGis e o GDAL.

O WinGis utiliza esta biblioteca para leitura do arquivo shp, recebendo o nome da camada como retorno através do método infoShp.

A estrutura do método openshp() utilizada no WinGis para abrir um shapefile é representada na **Figura 1**.

## FUNDAMENTOS MATEMÁTICOS

Para a visualização do mapa na tela do computador foi necessário mapear o sistema de projeção dos mapas ao sistema de coordenadas da tela do computador. As coordenadas de Y no plano geográfico aumentam no sentido oposto de um plano cartesiano normal. Para correlacionarmos as coordenadas de tela em coordenadas de mapas plano retangulares utilizamos as seguintes relações:

```

178 void TFShape::openshp()
179 {
180
181     String fname;
182     char sType [15]= "";
183     int ix;
184     int i, j, k, m, aa, hh;
185     SHPObject *psShape;
186     int iPart, bValidate = 0, nInvalidCount=0;
187     int x, y;
188     int ii = 0;
189
190     xhSHP = SHPOpen( strfile.c_str(), "r+b" );
191
192     if ( xhSHP != NULL ) { ...
315         } // if ghSHP
316
317     defscaler();
318
319     } // openshp
320
321     int TFShape::xwin( double x ) { ...
324     } // xwin

```

*Figura 1. Método openshp() da biblioteca shapelib*

$$X_b = X_{\min} + P_x * (X_{\max} - X_{\min}) \quad (1)$$

$$Y_b = Y_{\min} + P_y * (Y_{\max} - Y_{\min}) \quad (2)$$

Onde:

(X<sub>b</sub>, Y<sub>b</sub>) representam as coordenadas de um ponto no mapa;

X<sub>min</sub> é o menor valor do eixo X do mapa;

Y<sub>min</sub> é o menor valor do eixo Y do mapa;

e P<sub>x</sub> e P<sub>y</sub> são dados por:

$$P_x = (X_a - X_{T\min}) / (X_{T\max} - X_{T\min}) \quad (3)$$

$$P_y = (Y_a - Y_{Tmin}) / (Y_{Tmax} - Y_{Tmin}) \quad (4)$$

Onde:

(X<sub>a</sub>, Y<sub>a</sub>) representam as coordenadas de um ponto na tela;

X<sub>Tmin</sub> é o menor valor do eixo X da tela;

Y<sub>Tmin</sub> é o menor valor do eixo Y da tela;

## MAPSERVER

O MapServer é um WMS (Web Map Service) gratuito e multiplataforma que permite a visualização de mapas geográficos conforme as requisições do sistema GIS. É utilizado em uma aplicação da NASA chamado TerraSIP.

O MapServer carrega o arquivo \*.map que é formatado pela aplicação GIS. Este arquivo possui a seguinte estrutura (**Figura 2**).

## QT

O Qt é um framework multiplataforma, gratuito e open source. Foi desenvolvido pela empresa Trolltech e atualmente pertence a NOKIA. Este software permite a criação de aplicações e bibliotecas em C++, onde é possível compilá-las para diversos sistemas operacionais sem ter que fazer alterações no código do programa.

A arquitetura Model/View integrada com um ambiente visual (QtCreator) auxiliam o desenvolvedor a localizar e criar rapidamente vários tipos de objetos, dando maior produtividade ao programador. O QtCreator é uma interface visual que integra alguns módulos auxiliares ao QT. Entre eles:

- QtDesign – Ambiente visual para construção de interfaces gráficas e integração com o código fonte através de SIGNAL/SLOTS. Este

```

1 MAP
2
3 EXTENT -51.0258 -18.7623 -49.735 -16.4574
4 SHAPEPATH './mapas'
5 SIZE 571 351
6
7
8 LAYER
9
10 NAME 'C:/WinGis/mapas/go.shp'
11 TYPE POLYGON
12 DATA 'C:/WinGis/mapas/go.shp'
13 STATUS DEFAULT
14
15
16 CLASS
17 NAME 'C:/WinGis/mapas/go.shp'
18 OUTLINECOLOR 222 222 222
19 COLOR 1403648329 -935983160 34207618
20
21 END
22 END
23 LAYER
24
25 NAME 'C:/WinGis/mapas/parques.shp'
26 TYPE POLYGON
27 DATA 'C:/WinGis/mapas/parques.shp'
28 STATUS DEFAULT
29
30
31 CLASS
32 NAME 'C:/WinGis/mapas/parques.shp'
33 OUTLINECOLOR 222 222 222
34 COLOR 170 0 0
35
36 END
37 END
38 END

```

→ Quadrante do Mapa  
 → Pasta com Shapes  
 → Tamanho do Janela

Informações da camada (go)

Informações da camada (Parques)

*Figura 2. Estrutura do arquivo map.*

procedimento economiza tempo, portanto gera maior produtividade decorrente da associação entre o código e a interface ser feita visualmente sem a necessidade de codificação.

- QtLinguist – Converte a sua aplicação para qualquer idioma sem a necessidade de redigitar o código.

- QtAssistant – Possui diversos exemplos de códigos fontes abertos, exemplificando suas funcionalidades.
- O Qt traz consigo um novo padrão de alta performance com bibliotecas OpenGL e Web nativas do software. A integração entre componentes visuais e o framework possibilitou a criação de uma aplicação voltada a análise de informações geográficas.

Outra funcionalidade inovadora do QT é a forma de conectar objetos através de SIGNALS e SLOTS, onde o signal é a informação transmitida pelo usuário e slots são os métodos a serem executados. São responsáveis pela interação entre os vários objetos da aplicação, conforme **Figura 3**.

## RESULTADOS

### WINGIS

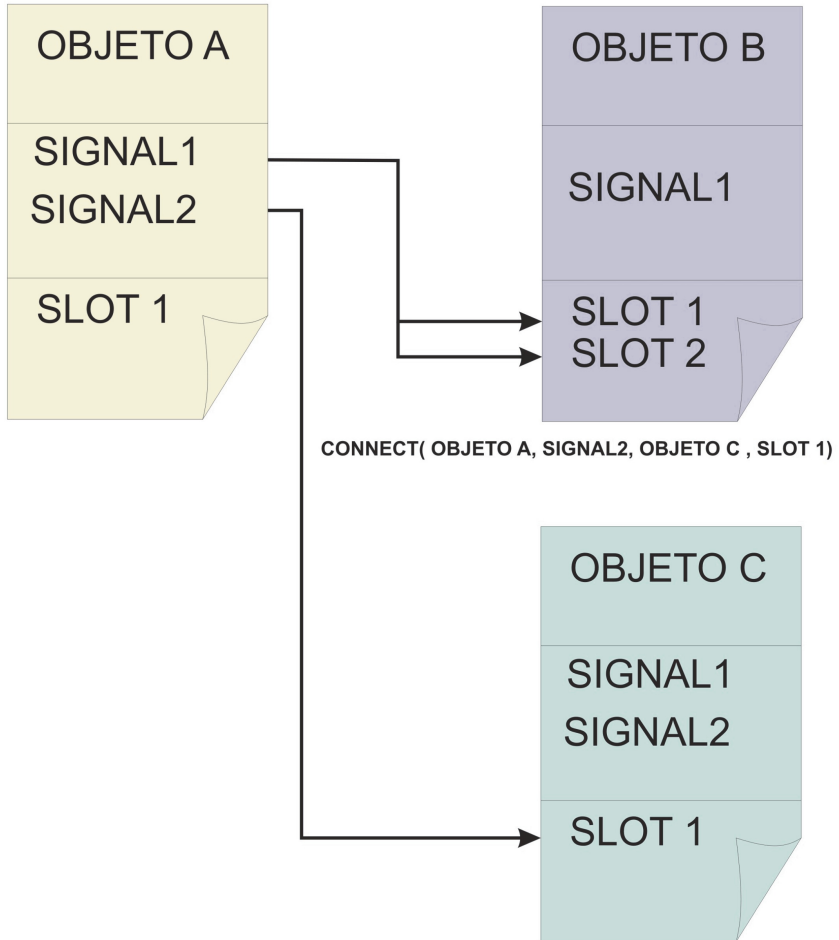
O Wingis é um Sistema de Informações Geográficas (SIG), baseado no conceito de que um SIG é um software que permite a captura, modelagem, manipulação, recuperação, análise e apresentação de dados georreferenciados (WORBOIS, 1995).

O WinGis é resultado do estudo realizado com a ferramenta QT e aplicado na análise e visualização de shapefiles em camadas (**Figura 4**). O WinGis foi desenvolvido de forma que integrasse diversos repositórios de dados geográficos com a alta performance da programação C.

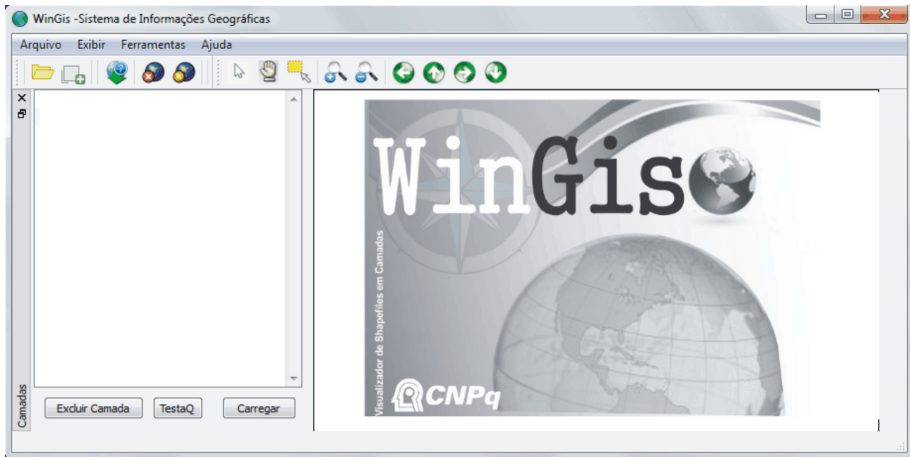
Considerando que as aplicações voltadas para a análise de imagens, tais como ESRI ArcIMS e Intergraph GeoMedia WebMap, na maioria são sistemas proprietários, podemos considerar que o uso do QT é vantajoso devido a isen-



CONNECT( OBJETO A, SIGNAL1, OBJETO B , SLOT 1)  
CONNECT( OBJETO A, SIGNAL1, OBJETO B , SLOT 2)



*Figura 3. Interação dos Objetos utilizando Signals e Slots.*

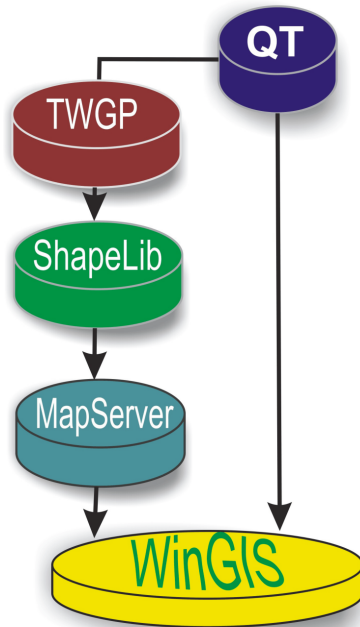


*Figura 4. Interface inicial do WinGis.*

ção do custo de licenciamento. Avaliando os recursos técnicos do QT, pudemos perceber que o ambiente GUI se destaca por possuir diversos recursos visuais que auxiliam no desenvolvimento e agilizam a programação.

O WinGis faz uso da biblioteca shapelib e MapServer para construção dos mapas. Estas bibliotecas interagem com a classe TWGP que informa os parâmetros de construção do mapa. (**Figura 5**).

Por ser um sistema de apoio à decisão que envolve a integração de dados espacialmente referenciados, em um ambiente para resolução de problemas (COWEN, 1988) o WinGis é um sistema completo para visualização de shapefiles dispostos em camadas sobrepostas.



*Figura 5. Arquitetura do WinGis.*

## ARQUIVOS DE PROJETO WGP

Para a visualização dos arquivos vetoriais foi desenvolvida uma classe de tratamento de mapas chamada TWGP. Esta classe permite definir a estrutura e formato que o mapa será apresentado no WinGis.

O arquivo de projeto gerado pelo WinGis (com extensão wgp) é baseado em uma estrutura simples e contém as informações necessárias para a integração de diferentes informações espaciais de um dado projeto (**Figura 6**).

```

1 PROJETO_WINGIS
2 DIRETORIO
3 ../mapas → Diretórios com os Shapes
4 JANELA
5 571
6 331 → Tamanho da Janela do mapa
7 QUADRANTE
8 -180
9 -90
10 180
11 90 → Referência de posicionamento
12 SHAPES
13 2 → Quantidade de Camadas
14 malhaGO
15 brasil → Nome das Camadas
16 0
17 1 → Ordenação das Camadas
18 1
19 1 → Visibilidade das Camadas

```

**Figura 6.** Estrutura do arquivo WGP.

A classe responsável por gerar o arquivo wgp é a TWGP (**Figura 7**). Esta classe possui todos os métodos responsáveis pelas requisições ao MapServer e a Shapelib.

Os componentes auxiliares utilizados na geração dos arquivos TWGP é a biblioteca shapelib e o MapServer podendo ser considerados como parte da estrutura do WinGis. Estas bibliotecas utilizadas também são softwares livres.

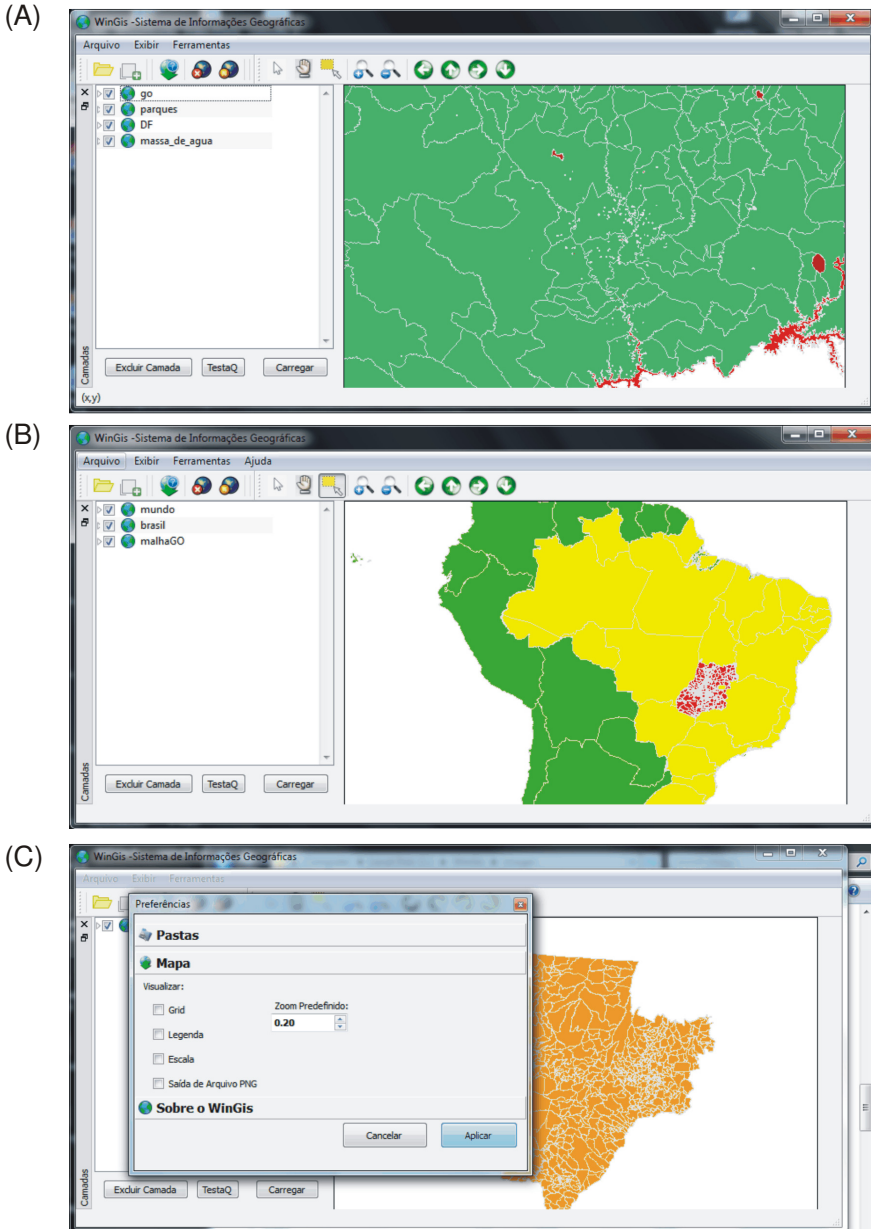
TWGP
<ul style="list-style-type: none"> <li>- Diretorio : QString</li> <li>- ArqShapes : QString</li> <li>- Tipo[50] : int</li> <li>- Visivel[50] : int</li> <li>- Ordem[50] : int</li> <li>- QtArqShapes : int</li> <li>- Quadrante[4] : double</li> <li>- QuadShapes[50][4] : double</li> <li>- CorLinhaShapes[50][3] : int</li> <li>- CorFundoShapes[50][3] : int</li> <li>- TamJanela[2] : int</li> <li>- Legenda : bool</li> <li>- Escala : bool</li> <li>- Saida : bool</li> </ul>
<ul style="list-style-type: none"> <li>+ IncluirCamadaSHP(QString pnomarq) : boolean</li> <li>+ ExcluirCamadaSHP(int pindice) : boolean</li> <li>+ SalvaProjeto(QString pnomarqwgwp) : boolean</li> <li>+ CarregaProjeto(QString pnomarqwgwp) : boolean</li> <li>+ ExportaMAP(QString pnomarqmap, bool pLegenda, bool pEscala, bool pSaida, bool pGrid) : boolean</li> <li>+ GetNomeCamada(int pcamada) : QString</li> <li>+ Atualiza_Quadrante() : void</li> <li>+ Zoom(double pfator) : void</li> <li>+ Enquadramento(double *Janela) : void</li> <li>+ Geografico(QPoint *pxy) : void</li> <li>+ PontoGeografico() : QPointF</li> <li>+ Move(double pfator, int ptipo) : void</li> <li>+ MoveX(double pMovX, double pMovY) : void</li> <li>+ SelZoom(double Xini, double Yini, double XFim, double YFim) : void</li> <li>+ InsereCores(int pindice) : boolean</li> <li>+ TWGP() : void</li> <li>+ TWGP(QString pdiretorio, int pJanLargura, int pJanAltura) : void</li> </ul>

*Figura 7. Classe TWGP.*

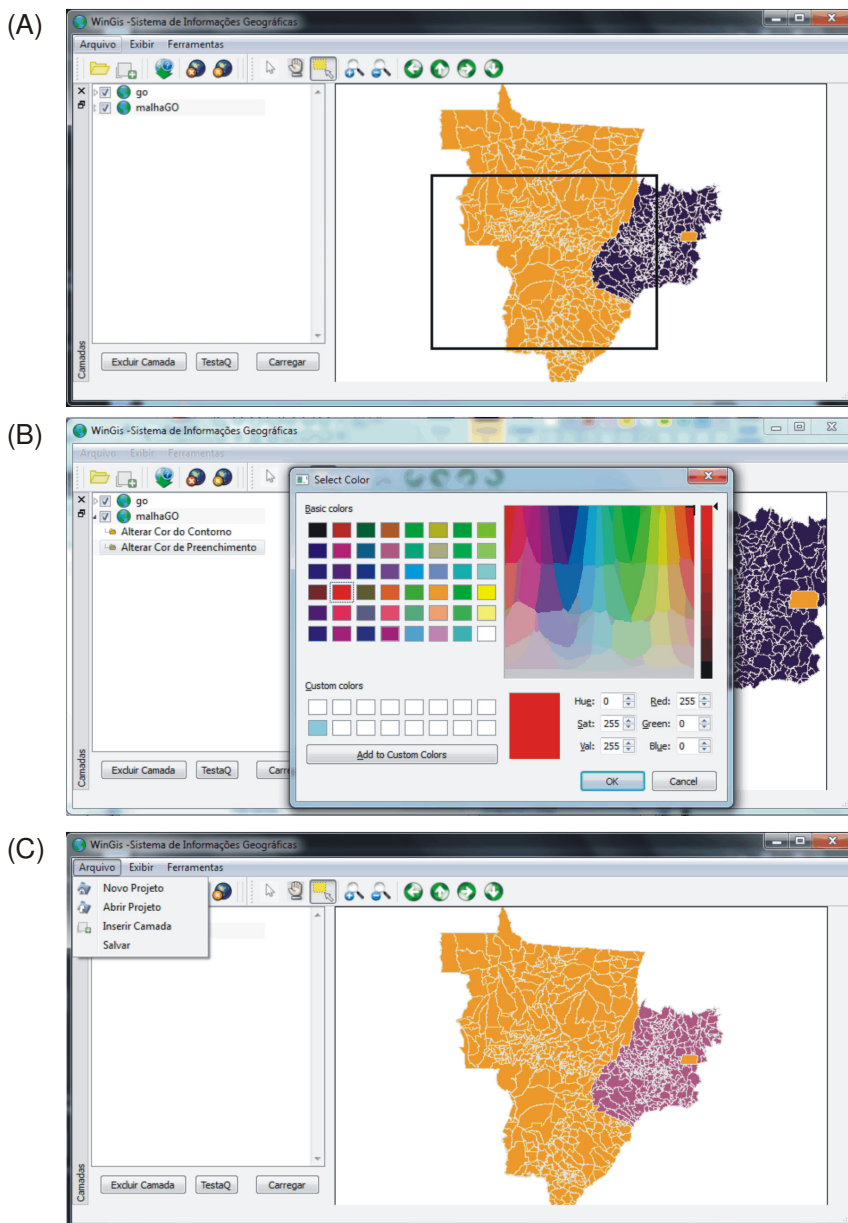
As **Figuras 8 e 9** retratam as principais funcionalidades e visualizações do protótipo WinGis.

## CONCLUSÃO

O ambiente QT de programação orientada a objetos e interfaceada por uma aplicação GUI promoveram uma maior velocidade à fase de desenvolvimento das interfaces e das funcionalidades de manipulação de arquivos vetoriais e rasters do WinGis. A possibilidade de integração com o servidor de mapas é



**Figura 8.** Interfaces do programa WinGIS: (a) visualização de shapefiles apresentados em camadas; (b) visualização de shapefiles apresentados em camadas; e (c) interface das preferências referentes ao mapa.



**Figura 9.** Interfaces do programa WinGIS: (a) zoom por seleção; (b) seleção das cores de preenchimento e contorno das camadas; e (c) Menu Arquivo.

outro diferencial que viabilizou ao WinGis ter acesso às funcionalidades do MapServer.

O WinGis se configurou em um ambiente amigável para a visualização integrada e em camadas de dados geográficos do tipo vetorial e raster. E todas as ferramentas e componentes utilizadas no seu desenvolvimento são gratuitas. Suas interfaces permitem a visualização conjunta de imagens de satélites, pontos, linhas e polígonos de uma forma fácil de operação.

### **AGRADECIMENTOS**

Agradecemos ao departamento de Computação do Centro Universitário de Anápolis – UniEVANGELICA que forneceu o ambiente da Fábrica de Tecnologias Turing (FTT) como suporte tecnológico para o desenvolvimento das pesquisas e do sistema WinGis. Agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro ao discente pesquisador em forma de bolsa de iniciação científica durante todo o período da pesquisa.

### **REFERÊNCIAS BIBLIOGRÁFICAS**

COWEN D. J. (1988). SIG versus CAD versus DBMS: what are the differences?, em “Introductory readings in Geographic Information Systems”. Londres: Taylor and Francis.

INSTITUTO DE PESQUISA DOS SISTEMAS AMBIENTAIS, Inc. (Julho, 1998). “Descrição técnica de ESRI Shapefile”. Recuperado sobre 2007-07-04.

PAREDES, E. A. (1994). Sistema de Informação Geográfica. 1ª ed. São Paulo: Érica, 95p.

ROSA (1992) Introdução ao Sensoriamento Remoto, Uberlândia: Editora da UFU, 109p.



TEIXEIRA, A. L. de A.; MORETTI, E.; CHRISTOFOLETTI, A. Introdução aos sistemas de informação geográfica. Edição do Autor, Rio Claro, 1992. 80 p.

UNIFAP. Cartografia Básica. Disponível em: <<http://www.cartografia.eng.br/cartografia/artigos/ncarto01.php>> Acesso em: 11 dez.2009.

UNIVERSIDADE ESTADUAL PAULISTA (Brasil). SIG no Ensino de Geografia. Disponível em: <<http://orbita.starmedia.com/~ensinogeosig/index3.html>> Acesso em: 25 fev. 2010.

WORBOIS, Michael F. (1995) GIS: A Computing Perspective. Londres: Taylor and Francis.