

## O Formato 'Shapefile' como Representação de Dados

Prof. MSc. Henrique Firkowski <sup>1</sup>

Prof. Dr. Carlos A. Picanço de Carvalho <sup>2</sup>

MSc. Luciene Stamato Delazari Skroch <sup>3</sup>

Eng. Cart. Maria de Lourdes A. M. Gonçalves <sup>4</sup>

UFPR- Departamento de Geomática  
Curso de Pós-Graduação em Ciências Geodésicas  
Caixa Postal 19.001  
Curitiba – PR

<sup>1</sup> ✉ [firk@geoc.ufpr.br](mailto:firk@geoc.ufpr.br)

<sup>2</sup> ✉ [carlosac@inf.ufpr.br](mailto:carlosac@inf.ufpr.br)

<sup>3</sup> ✉ [luciene@geoc.ufpr.br](mailto:luciene@geoc.ufpr.br)

<sup>3</sup> ✉ [mlaquino@geoc.ufpr.br](mailto:mlaquino@geoc.ufpr.br)

<b>Conteúdo</b>	<b>1 Introdução</b> <b>2 O Formato Shapefile</b> <b>3 Estrutura Genérica para Manipulação de Shapefiles</b> <b>4 Considerações sobre o Formato Shapefile como Representação de Dados Cartográficos</b> <b>5 Referências Bibliográficas</b>
-----------------	--

**Resumo:** No presente trabalho faz-se uma abordagem do formato de dados shapefile, proposto como formato padrão do conjunto de classes denominado MapObjects 2.0. Este produto constitui-se num conjunto de classes que permite visualizar e operar dados presentes em shapefiles. Um shapefile armazena tanto dados de geometria quanto dados de atributo. A estrutura dos diversos tipos de shapefiles é descrita neste trabalho a partir das suas especificações e do desenvolvimento de um aplicativo Windows escrito em Linguagem Microsoft Visual C++. Neste aplicativo desenvolveu-se, a partir dos conceitos da linguagem C, um conjunto de estruturas que permite conter dados de qualquer tipo de shapefile, e realizar operações de acesso a dados de geometria e de atributo. Foram avaliadas as limitações e potencialidades dos shapefiles e sua adoção como estrutura de dados. A necessidade do entendimento da organização dos shapefiles é parte de um processo maior de desenvolvimento de um software para tratamento e visualização de informações espacial em andamento no Curso de Pós-Graduação em Ciências Geodésicas da Universidade Federal do Paraná.

**Palavras chave:** Cartografia, estrutura de dados, shapefile.

**Abstract:** This paper presents a shapefile format description, proposed as standard for MapObjects 2.0. This product is a set of classes that allows operating and visualizing shapefile data. Shapefile stores both geometric and attribute data. Several shapefile structures and a Windows application written in Microsoft Visual C++ are described. For allowing access and for storing any kind of shapefile data, it was developed a general data structure based on C language concepts. It is made an evaluation of the shapefile format for cartographic applications. In the graduate program on Geodetic Sciences at Federal University of Paraná, Brazil is in course a project for developing software for treatment and visualization of spatial information using the knowledge in shapefile.

**Keywords:** Cartography, data structure, shapefile.

### 1 Introdução

A cartografia digital consiste na representação e manipulação de dados em ambiente computacional (Clarke, 1990). Todo aplicativo para cartografia digital apresenta uma forma de representação de dados, deve permitir a realização de operações com dados, e finalmente deve possibilitar a visualização destes dados.

Os dados cartográficos utilizados para descrever objetos do mundo real têm três componentes: posicional, de atributo, e de relacionamentos. Normalmente, o cartógrafo faz uma abstração destes objetos e os representa por meio de pontos, linhas e polígonos. Cada elemento possui então, uma componente espacial ou geométrica descrita por coordenadas bi ou tridimensionais. Associadas a estas posições existem atributos que descrevem ocorrências ou características dos elementos e, além disso, podem existir relacionamentos entre os diversos elementos espaciais (topologia).

O modelo de dados descrito acima é denominado modelo de dados vetorial, em que os objetos do mundo real são representados por coordenadas retangulares definidas de tal modo a descrever o objeto com uma acurácia adequada ao propósito da representação. Outro modelo de dados conceitual é o modelo matricial, em que a unidade de representação é uma unidade de área dentro de uma grade com um valor associado. A ocorrência de repetição do valor associado indica um mesmo objeto. Burrough (1998), acrescenta aos modelos conceituais o modelo baseado em objetos e o modelo baseado em campo. No primeiro, o mundo é visto como constituído de objetos que tem suas descrições particulares e existe uma hierarquia envolvida no seu estabelecimento. No segundo modelo, o mundo é visto como sendo constituído de fatos/fenômenos que podem ser descritos por funções matemáticas contínuas.

Em arquivos de computador, normalmente, diferentes organizações ou formatos de dados são identificadas pelo sufixo do nome do arquivo. Dentre os diversos formatos existentes os mais conhecidos são os formatos DWG (AutoCAD), DGN (MicroStation), e DXF que é adotado para troca de dados entre programas. O formato *shapefile*, desenvolvido pela ESRI (Environmental Systems Research Institute), é mais um destes formatos de dados apropriado para armazenar tanto geometria quanto atributos de dados cartográficos.

Neste trabalho descreve-se o formato *shapefile*, apresentando os tipos de dados possíveis como representação, um aplicativo desenvolvido para realizar operações de acesso aos dados e comenta-se a aplicabilidade deste formato de dados para representar dados cartográficos. A parte experimental do tema que trata este trabalho foi desenvolvida usando a linguagem de programação Microsoft Visual C++ e o MapObjects 2.0.

## 2 O Formato Shapefile

O *shapefile* é uma organização de dados idealizada pela empresa ESRI para conter tanto dados geométricos quanto dados de atributos. Os dados armazenados segundo o formato *shapefile* estão distribuídos em três arquivos. Dois deles contém os dados propriamente ditos, e um terceiro contém informação do índice de organização dos dados presentes nos primeiros.

O formato de armazenamento *shapefile*, admite cinco tipos básicos de dados: ponto, polilinha, polígono, multiponto e *multipartch*. Estes tipos sofrem variações para armazenar além das coordenadas *xy*, as coordenadas *xym* e *xyzm*, perfazendo 14 tipos (Tabela 1).

Tabela 1 - Tipos de dados dos possíveis em *Shapefile*

Tipo/conteúdo	xy	xyzm	xym
ponto	Ponto (1)	PontoZ (11)	PontoM (21)
linha	Polilinha (3)	PolilinhaZ (13)	PolilinhaM (23)
área	Polígono (5)	PolígonoZ (15)	PolígonoM (25)
ponto	Multiponto (8)	MultipontoZ (18)	MultipontoM (28)
área	---	Multipartch (31)	---

Na tabela acima estão reunidos os tipos possíveis de armazenamento em *shapefiles*. Nos tipos 1, 3, 5, 8 são armazenadas coordenadas retangulares (*x,y*) para pontos, polilinhas, polígonos e conjuntos de pontos. Para os tipos 11, 13, 15, 18 e 31 são armazenados pontos, linhas, áreas, conjuntos de pontos e figuras complexas, quando são dadas as coordenadas (*x,y,z,m*). E para o conjunto de tipos 21, 23, 25 e 28 as coordenadas são (*x,y,m*). A coordenada *m*, denominada *measure*, indica a posição de um atributo em relação a posição (*x,y,z*), ou (*x,y*).

Os três arquivos de um conjunto *shapefile* tem o mesmo nome e são diferenciados por suas extensões. Os dados geométricos são armazenados em arquivos com extensão *.shp*, denominado arquivo principal. Os dados de atributos, com extensão *.dbf*, são armazenados em arquivos dBASE (dBASE, Inc.) e podem ser criados/editados com este programa. O terceiro arquivo, com extensão *.shx*, contém o endereço e o tamanho de cada registro do arquivo principal (*.shp*). A Figura 1 apresenta um conjunto *shapefile*.

Cada conjunto *shapefile* só admite um tipo de dados, isso significa que deve ser um dos tipos descritos na Tabela 1. Uma mesma visualização pode ser composta por mais de um conjunto *shapefile*. Desse modo uma representação com limites administrativos, hidrografia, rede viária implicará a existência de pelo menos três conjuntos de *shapefiles*, um para cada tipo de feição.

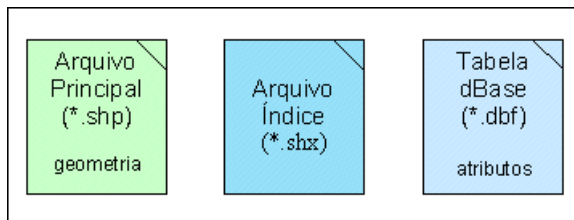


Fig. 1 : Conjunto *shapefile*

Os cabeçalhos dos arquivos *.shp* e *.shx* têm 100 bytes. Nestes estão contidas as informações: código do arquivo, valor numérico do tamanho do arquivo em palavras de 16 bytes, versão do formato, tipo de dados e coordenadas *xyzm* extremas. A diferença entre os cabeçalhos *.shp* e *.shx* está apenas no valor numérico do tamanho do arquivo. O cabeçalho de um arquivo *.dbf* é formado por duas partes, uma fixa e outra variável. Na parte fixa, com comprimento de 32 bytes, estão os dados de: data de atualização do arquivo, número de registros na tabela, número de bytes do cabeçalho e número de registro. A parte variável do cabeçalho é composta por múltiplos de 32 bytes, um para cada atributo. Assim, se houver 4 atributos haverá 4x32 bytes na parte variável do cabeçalho. Os atributos podem ser variáveis do tipo: carácter, lógica, data, numérica, e texto (string).

Num conjunto *shapefile* existe uma correspondência unívoca entre os registros de dados dos três arquivos. Isto é, para o primeiro registro de dados do arquivo *.shp* existe em correspondência um registro no arquivo de índices (*.shx*), e um registro no arquivo de atributos (*.dbf*), como se mostra na Figura 2.

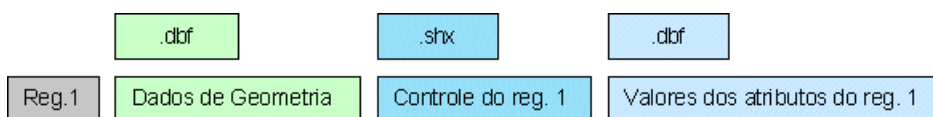


Fig. 2 : Relacionamento entre registros dos *shapefile*

## 3 Estrutura Genérica para Manipulação de Shapefiles

Para adquirir experiência acerca da organização dos dados em *shapefiles* foi desenvolvido um aplicativo Windows em Linguagem Microsoft Visual C/C++ denominado *LayerEDT*. Este aplicativo passou por aprimoramentos que conduziram ao estabelecimento de uma estrutura genérica para manipulação destes dados.

Neste aplicativo é possível abrir *shapefiles* e visualizar de modo alfanumérico todos os dados de geometria e de atributos que os

compõem. Foram desenvolvidos diálogos e funções para realizar operações de edição de valores geométricos das feições, e valores alfanuméricos dos atributos. Podem ser adicionados ou removidos tanto registros de dados geométricos quanto variáveis de atributo. Com este aplicativo, um *shapefile* pode ser criado manualmente a partir de uma lista de coordenadas e de valores de seus atributos.

Foram usados vários recursos da Linguagem Microsoft Visual C/C++, como por exemplo o tipo de dados *struct*, e as classes de *collections*. Com estes tipos de dados e com as classes destinadas à manipulação de coleções foi construída uma estrutura genérica. Para armazenar os dados geométricos dos arquivos .shp foram criadas algumas estruturas. A primeira delas é a estrutura *Ponto*, que prevê variáveis do tipo *double* para coordenadas *x*, *y*, *z* e valores *m*. Esta estrutura é utilizada por outras estruturas mais gerais.

Os dados geométricos de qualquer *shapefile* podem ser armazenados usando uma estrutura chamada *str\_Elemento*. Dependendo do tipo de dado presente no *shapefile*, se ponto, linha, polígono, multiponto ou multipatch, utilizam-se todos ou parte dos dados previstos na estrutura. Em geral os *shapefiles* são compostos por mais de um registro e então para que se possa armazenar todos os registros deve existir um conjunto de *str\_Elemento*. A administração deste conjunto é feita por um objeto *CList* denominado *L\_Elementos*.

Em termos práticos, o armazenamento de dados provenientes de *shapefiles* ocorre do seguinte modo: a) uma estrutura do tipo *str\_Elementos* é definida; b) os valores de um registro são lidos para os variáveis correspondentes de acordo com o tipo (*shapefile*); e c) toda a estrutura é passada para o objeto *L\_Elementos*.

Cada valor presente na lista de atributos do registro tem uma especificação armazenada na estrutura *str\_dbf*. Nesta estrutura estão declaradas as variáveis da especificação dos campos de um arquivo no formato dBASE (.dbf). Os valores dos atributos também são introduzidos no objeto *L\_Elementos*.

A organização dos dados de *shapefiles* neste aplicativo pode ser vista na Figura 3. A recuperação dos dados do *shapefile* desta estrutura estabelecida é relativamente simples e permite que sejam feitas diversas operações geométricas, como por exemplo: determinação de coordenadas extremas, cálculo de distâncias, áreas, volume, ângulos de orientação, dentre muitas outras. Do mesmo modo podem ser realizadas operações de recuperação e uso de valores de atributos. Na Figura 4 apresenta-se a tela inicial do aplicativo *LayerEDT* que faz uso da estrutura mostrada na Figura 3. Nesta tela é apresentado o conteúdo de um *shapefile*, onde a coluna *TIPO* representa o tipo de dado do arquivo e as demais colunas representam os atributos associados aos dados geométricos. Quando se clica sobre o dados ou textos desta janela são abertos diálogos para realizar a edição de dados, edição de atributos, inserção/remoção de registro ou edição de especificações de atributo.

#### 4 Considerações sobre o Formato *Shapefile* como Representação de Dados Cartográficos

O uso de uma representação de dados em cartografia é caracterizado pelo acesso aos seus elementos métricos e aos seus elementos semânticos. Portanto, os dados de cartografia devem sofrer ações de busca métrica e/ou semântica e também ações de renderização. Uma extensão do uso dos dados de cartografia está na geração de buscas condicionadas por injunções topológicas. Para isso a base de dados deve ter uma organização que contemple a representação da topologia, e o formato *shapefile* não contempla esta representação. O uso dos dados fica restrito ao uso convencional de dados em cartografia digital.

Em cartografia as feições do mundo real são representadas utilizando símbolos pontuais, lineares e de área, além de textos. Estas representações estão sujeitas a variações em forma, dimensão, orientação e cor. Como abordado no Ítem 2, o formato *shapefile* foi concebido para o armazenamento de dados no formato vetorial, ou seja, pontos, linhas e polígonos e outras estruturas mais complexas.

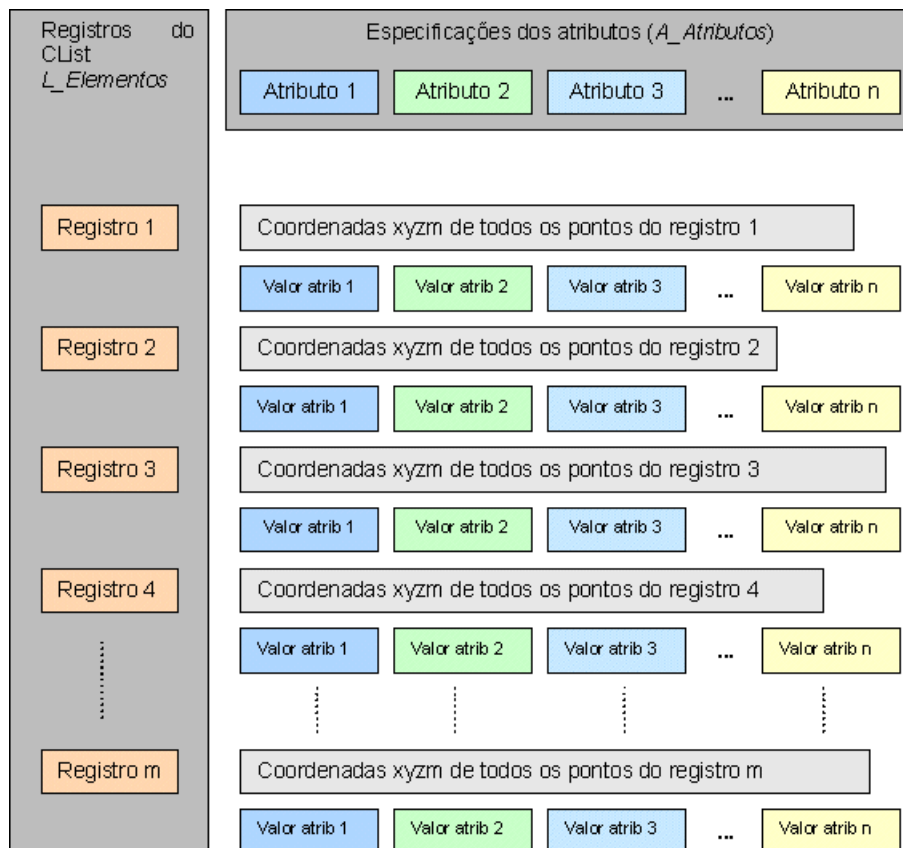


Fig. 3 : Organização dos dados de um *shapefile* no aplicativo *LayerEDT*

Acerca deste formato para os objetivos de representação de dados cartográficos pode-se dizer:

- o formato *shapefile* permite a representação de dados de geometria (posições ou coordenadas), sem restrições;
- a organização dos dados em arquivos diferentes, de acordo com o tipo de feição que está sendo representada (ponto, linha ou polígono), é interessante do ponto de vista cartográfico, e está em acordo com conceito de 'nível de informação';
- as variações de forma, dimensão, cor e orientação são possíveis utilizando uma linguagem de programação e outras classes/objetos do MapObjects 2.0; e
- esta organização não permite armazenar topologia.

Visto que o MapObjects 2.0 consiste de um conjunto de classes com uma hierarquia estabelecida, existe possibilidade de acessar métodos e propriedades e adequar a simbologia aos propósitos da visualização que se almeja. A simbolização dos elementos a serem visualizados em um aplicativo com funcionalidades do MapObjects 2.0 é feita internamente usando objetos de suas próprias classes.

O interesse pelo uso de *shapefiles* está na possibilidade de desenvolver protótipos e aplicativos para cartografia, como área de interesse do grupo de Cartografia e Sistemas de Informação Geográfica do Curso de Pós-Graduação em Ciências Geodésicas da Universidade Federal do Paraná. Alguns experimentos em desenvolvimento já foram realizados; um destes foi apresentado neste evento no trabalho intitulado "MapObjects 2.0 – Uma Ferramenta para Desenvolvimento de Protótipos e Aplicativos para Cartografia".

REG.	TIPO	AREA	PERIMETER	MUNICIP_	MUNICIP_ID	CIDADE	POPULACAO
1	Poligono	0.010576	0.428266	2	1	JARDIM OLINDA	1411
2	Poligono	0.013767	0.604407	3	2	ITAGUAJE	5054
3	Poligono	0.019282	0.609045	4	3	DIAMANTE DO NORTE	7602
4	Poligono	0.057630	1.027530	5	5	TERRA RICA	13905
5	Poligono	0.019574	0.629942	6	4	PARANAPOEMA	2452
6	Poligono	0.014159	0.530471	7	6	SANTA INES	2046
7	Poligono	0.026721	0.968256	8	7	NOVA LONDRINA	12848
8	Poligono	0.111461	1.675510	9	9	PARANAVAI	71173
9	Poligono	0.021236	0.594953	10	10	SANTO ANTONIO DO CAIUA	3112
10	Poligono	0.025852	0.708521	11	8	SANTO INACIO	5514
11	Poligono	0.020057	0.601146	12	12	MARILENA	6677
12	Poligono	0.016550	0.559553	13	11	INAJA	2642
13	Poligono	0.027245	0.704921	14	13	PORECATU	17103
14	Poligono	0.011632	0.450225	15	14	LUPIONOPOLIS	4458

**Fig. 4 :** Tela do aplicativo *LayerEDT* com dados de um *shapefile*  
 Fonte de dados: Centro Integrado de Estudos em Geoprocessamento – CIEG/UFPR

## 5 Referências Bibliográficas

Burrough, P.A.; McDonnel, R.A. *Principles of Geographical Information Systems*. Oxford University Press, Somerset, 1998.

Clarke, K.C. *Analytical and Computer Cartography*. Prentice Hall, Englewood Cliffs, 1990.

ESRI - *Building Applications with MapObjects*. 1999.

\_\_\_\_ - *MapObjects 2.0 – Programmer's Reference*. 1999.

\_\_\_\_ - *Shapefile Technical Description. An ESRI White Paper*. Julho 1998.

Firkowski, H., Carvalho, C.A.P., Skroch, L.S.D., Araki, H., Aquino, M.L. *GeoClasses – Base Para a Construção de um Software para Tratamento e Visualização de Dados Espaciais e Informações Georreferenciadas*. In I Colóquio Brasileiro de Ciências Geodésicas. Curitiba, 1999.

Firkowski, H., Carvalho, C.A.P., Skroch, L.S.D., Araki, H., Aquino, M.L. *Estruturação de dados em SHAPEFILES*. In I Colóquio Brasileiro de Ciências Geodésicas. Curitiba, 1999.

Firkowski, H. *Uma Base Computacional para Estudo de Generalização*. Seminário apresentado ao Curso de Pós-Graduação em Ciências Geodésicas da Universidade Federal do Paraná. Curitiba – maio de 2000.

Kraak, M.J. Ormeling, F.J. *CARTOGRAPHY – Visualization of Spatial Data*. Addison Wesley Longman Limited, Singapore, 1998.

Skroch, L. S. D. *A Utilização do MapObjects 2.0 para visualização cartográfica*. Seminário apresentado ao Curso de Pós-Graduação em Engenharia de Transportes da Universidade de São Paulo. São Paulo – maio de 2000.