UNIVERSIDADE DE SÃO PAULO – USP
Faculdade de Filosofia, Letras e Ciências Humanas
Departamento de Geografia
Programa de Pós-Graduação em Geografia Humana

RICARDO SALEIMEN NADER

**INTEGRAÇÃO DE DADOS EM SIG:
UMA ABORDAGEM POR TECNOLOGIAS DE USO LIVRE**

Versão corrigida

São Paulo
2015

RICARDO SALEIMEN NADER

**Integração de dados em SIG:
uma abordagem por tecnologias de uso livre**

Tese apresentada à Faculdade de Filosofia,
Letras e Ciências Humanas do
Departamento de Geografia da Universidade
de São Paulo para a obtenção do título de
doutor em Geografia Humana.

Área de Concentração: Geografia Humana

Orientador: Prof. Dr. Reinaldo Paul Perez
Machado.

São Paulo
2015

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo na Publicação

Serviço de Bibliotecas e Documentação

Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo

N135i Nader, Ricardo Saleimen Integração de dados em
SIG: uma abordagem por tecnologias de uso livre /
Ricardo Saleimen Nader ; orientador Reinaldo Paul
Pérez Machado. - São Paulo, 2015.
139 f.

Tese (Doutorado)- Faculdade de Filosofia, Letras
e Ciências Humanas da Universidade de São Paulo.
Departamento de Geografia. Área de concentração:
Geografia Humana.

1. SISTEMA DE INFORMAÇÃO GEOGRÁFICA. 2. CARTOGRA-
FIA. 3. MAPAS. 4. GEOGRAFIA. I. Machado, Reinaldo
Paul Pérez, orient. II. Título.

FOLHA DE APROVAÇÃO

NADER, Ricardo Saleimen.

INTEGRAÇÃO DE DADOS EM SIG: UMA ABORDAGEM POR TECNOLOGIAS DE
USO LIVRE

Tese apresentada à Faculdade de Filosofia,
Letras e Ciências Humanas do Departamento de
Geografia da Universidade de São Paulo para a
obtenção do título de doutor em Geografia
Humana

Aprovado em ____/____/____

Banca Examinadora:

Prof. Dr.

Julgamento: _____

Assinatura

Prof. Dr.

Julgamento: _____

Assinatura

Prof. Dr.

Julgamento: _____

Assinatura

Prof. Dr.

Julgamento: _____

Assinatura

Prof. Dr.

Julgamento: _____

Assinatura

DEDICATÓRIA

Ao Prof. Raffo (*in memoriam*) e para todos aqueles a quem esta tese possa ser útil.

AGRADECIMENTOS

Agradeço, em primeiro lugar, ao Prof. Raffo (*in memoriam*), a quem devo a escolha do tema e a oportunidade de desenvolvê-lo no programa de doutoramento da USP.

Agradeço ao Prof. Reinaldo, amigo de anos do Departamento de Geografia da USP, em quem me espelhei para seguir os caminhos profissional e acadêmico na disciplina de Geoprocessamento e, quem me acolheu como orientando após o falecimento do Prof. Raffo.

Agradeço ao amigo Danilo Rogerio de Sousa quem, em muito, auxiliou na formatação da tese segundo as normas acadêmicas, contribuindo para o estabelecimento de um estudo mais científico.

Agradeço ao meu irmão Beck Nader, Prof. de Engenharia de Minas da UFMG, quem me auxiliou com valiosas dicas antes da entrega da versão final da tese.

Agradeço à Cris, por seu carinho e por sempre me apoiar nas diversas fases da pesquisa.

Agradeço aos participantes da banca de qualificação e defesa, pelas contribuições sugeridas.

Enfim, agradeço aos que, direta ou indiretamente, contribuíram para a existência deste trabalho.

Resumo

NADER, R. S. **Integração de dados em SIG: uma abordagem por tecnologias de uso livre**. 2015. 139 f. Tese (Doutorado) - Departamento de Geografia da Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo, 2015.

Foi demonstrada a viabilidade de utilização da biblioteca GDAL/OGR, por meio do seu emprego em processos de conversão sintática entre formatos de dados matriciais e vetoriais controlados, amplamente utilizados nas práticas profissional e de pesquisa em Geografia, Cartografia e Geoprocessamento.

Os dados convertidos e processados foram analisados comparativamente em seus parâmetros geométricos, topológicos, alfanuméricos e de referência espacial, de modo a identificar se existiram perdas ou manutenção de informações nesses indicadores. No caso da existência de perdas, foram empregados recursos disponíveis na biblioteca para ajustes nos parâmetros indicados, de modo a tentar retornar ao *conteúdo de dados* dos arquivos originais.

Finalmente, foi apresentada uma metodologia para a superação de desafios de integração de dados geoespaciais em SIG, por meio do emprego de sistemas de uso livre no campo do Geoprocessamento.

Palavras-chave: GDAL/OGR. Interoperabilidade. Conversão de dados.

Abstract

NADER, R. S. **GIS data integration: a free software approach**. 2015. 139 pg. Doctorate Thesis (Ph.D.) – Department of Geography, University of São Paulo, 2015.

It was demonstrated the feasibility of using the GDAL / OGR library, through its usage in syntax conversion processes between raster and vector controlled data formats, widely used in professional practice and research in Geography, Cartography and Geomatics.

The translated data were processed and analyzed comparatively in its geometrical, topological, alphanumeric and spatial reference parameters in order to determine if there were any loss or maintenance information on such indicators. In the case of losses, library resources were used into the indicated parameters in order to try to return to the original data files content.

Finally, we presented a methodology for overcoming geospatial data integration challenges in GIS, through the use of free softwares in the field.

Keywords: GDAL/OGR. Interoperability. Data translation.

SUMÁRIO

<u>INTRODUÇÃO</u>	5
CAPÍTULO 1 - FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA	11
1.1 Representação computacional de dados geográficos	11
1.2 Integração de dados geoespaciais em SIG	14
1.3 O FOSS (Free and Open Source Software) e a GDAL/OGR	19
1.4 Biblioteca GDAL/OGR	21
CAPÍTULO 2 - DEFINIÇÕES GEOMÉTRICAS	30
2.1 Geometrias suportadas pelo <i>Simple Features Specification</i>	30
2.2 Geometrias não suportadas pelo <i>Simple Features Especification</i>	34
2.3 Relacionamentos topológicos entre feições geográficas	37
2.4 Sistema Geodésico de Referência	42
CAPÍTULO 3 – FORMATOS DE INTERCÂMBIO DE DADOS GEOESPACIAIS	45
3.1 ESRI Shapefile	45
3.2 MapInfo MIF/MID	48
3.3 AutoCAD DXF	50
3.4 Google KML	52
3.5 GML	53
3.6 Microstation DGN	54
3.7 GeoTIFF	55
3.8 ArcInfo Grid	56
CAPÍTULO 4 - PROCESSOS DE CONVERSÃO SINTÁTICA ENTRE FORMATOS DE DADOS GEOESPACIAIS	57
4.1 Leitura e Escrita: AutoCAD DXF para ESRI Shape, MapInfo MIF/MID, Google KML, GML e Microstation DGN.	58
4.2 Leitura e Escrita: ESRI Shape para AutoCAD DXF	107

<u>4.3 Leitura e Escrita: ArcInfo Grid para GeoTIFF</u>	112
CAPÍTULO 5 - INTEGRAÇÃO DA GDAL/OGR COM SISTEMA GERENCIADOR DE BANCO DE DADOS ESPACIAIS	119
<u>5.1 Virtual OGR e Spatialite: acesso a dezenas de formatos vetoriais</u>	121
<u>5.2 Armazenamento de dados matriciais no Spatialite pelas bibliotecas GDAL e Rasterlite</u>	126
<u>CONCLUSÃO</u>	131
<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	133

Introdução

Estima-se que mais de 85% dos dados produzidos pela humanidade possuam alguma referência à superfície do planeta, sendo considerados portanto como dados geográficos¹. (STILLWEL et al, 1999).

Esses dados (analógicos e digitais), estão codificados em formas diferenciadas como textos, gráficos, planilhas, bancos de dados e, sobretudo, mapas.

O mapa tem sido, historicamente, um dos produtos resultantes da atuação do geógrafo. Não é sem sentido, portanto, que o processo de criação de mapas acabe por nomear a própria ciência da *Geografia* (*Geo*, Terra; *Grafia*, representação); ou seja, a representação gráfica do planeta e seus fenômenos.

Na prática profissional e de pesquisa é recorrente a utilização de mapas e outros produtos cartográficos relacionados como ortofotos, imagens de satélites, modelos numéricos de terrenos (MNT), etc, previamente criados por outros institutos, pesquisas e projetos.

A heterogeneidade de fontes e dados é um desafio constante para a sua plena integração nos distintos SIG (Sistemas de Informação Geográfica). Sendo esta a principal tecnologia para a geração de mapas e demais informações espaciais atualmente, a questão de sua integração é tema recorrente não só para a Cartografia, mas também para a própria ciência geográfica, como fator importante ao desenvolvimento de suas pesquisas.

A diversidade de aplicações (voltadas para as áreas ambiental, de urbanismo, de defesa, de gerenciamento de infraestruturas elétricas e de gás, etc) deu origem a SIG com funções diferenciadas, o que determinou a heterogeneidade e a quantidade dos formatos² atuais, levando a um desafio de integração de dados. Um grande complicador a esta questão é o fato de muitos formatos serem proprietários

1 Dados geográficos são também conhecidos como dados geoespaciais.

2 Formatos são entendidos como as estruturas lógicas utilizadas para o armazenamento de dados geográficos digitais em computador. Cada formato é estruturado para gerenciar os diferentes recursos disponíveis no SIG originário, ou seja, funções como topologia, visualização tridimensional, tipos de geometrias empregadas, entre outras, são condicionadas pelo formato utilizado no arquivo.

(pertencentes, portanto, a diferentes instituições) e não estarem publicamente acessíveis e documentados.

Assim, tendo em vista a necessidade constante de entrada e saída de mapas e demais dados geoespaciais em SIG, estes começaram a apresentar recursos próprios para a conversão entre diferentes formatos, bem como para o processamento avançado neles, buscando minimizar os problemas (geométricos, topológicos, de referência geodésica, entre outros) nas importações / exportações de dados.

No universo *open source*, uma biblioteca³ digital vem se destacando por sua incorporação em diferentes projetos de SIG livres: a GDAL/OGR (*Geospatial Data Abstraction Library* e *Simple Features Library*), utilizada para a conversão e processamento de dados geográficos matriciais e vetoriais.

Apesar da popularização, não há um estudo sistemático sobre sua eficiência na integração de dados em SIG, que justificasse o amplo emprego nos desafios de conversão sintática ou mesmo no abandono de soluções proprietárias equivalentes. Sua adoção se dá, sobretudo, pela forma em que é desenvolvida (facilitando a integração nos projetos de sistemas), pela documentação disponível e pela ausência, no mesmo estágio de desenvolvimento, de outros aplicativos *open source* semelhantes.

A hipótese desta pesquisa é a de que a GDAL/OGR seria suficiente para os processos de conversão sintática de dados geoespaciais.

A construção de dados geoespaciais é uma atividade custosa, tanto do ponto de vista financeiro quanto operacional. Isto fica evidente quando são considerados os diversos fatores envolvidos em sua geração como aerolevantamentos, aquisição de imagens satelitais, equipes técnicas envolvidas e conjunto de *softwares* e equipamentos empregados. Nesse sentido, a possibilidade de utilização de mapas previamente gerados e que possam ser reaproveitados, é de grande importância e

³ Biblioteca é uma coleção de códigos de programas utilizada no desenvolvimento de outros *softwares*. Bibliotecas fornecem recursos pré-existentes a programas independentes, tornando-os mais modulares e fáceis de atualizar.

se coloca como fator recorrente nos campos da Geografia, Cartografia e Geoprocessamento.

Portanto, o objetivo central deste estudo é demonstrar a viabilidade de utilização da biblioteca GDAL/OGR nos processos de conversão de dados geoespaciais. O objetivo específico é desenvolver e apresentar, a geógrafos e demais profissionais que lidam com a Cartografia e o Geoprocessamento, uma metodologia para a superação de problemas de integração de bases de dados geográficos, por meio da utilização de sistemas de uso livre.

Para isso, utilizaremos comparações em dados geográficos vetoriais e matriciais (controlados), comumente utilizados nas práticas profissional e de pesquisa em Geografia, como o Mapa Digital da Cidade (MDC) do município de São Paulo; ortofotos oriundas da Emplasa e o MNT utilizado para a ortorretificação das imagens Ikonos, da Secretaria do Meio Ambiente de São Paulo.

Os dados geoespaciais vetoriais desta pesquisa, foram convertidos pela biblioteca GDAL/OGR (em sua implementação gráfica no QGIS; pelo aplicativo **ogr2ogr** na linha de comando e pela geração de código utilizando o recurso **OGR VRT**) nos formatos selecionados para a investigação, e analisados comparativamente em seus parâmetros geométricos, topológicos, alfanuméricos e de referência espacial, de modo a identificar se existiram perdas ou manutenção de informações nesses indicadores. No caso da existência de perdas, foram empregados recursos disponíveis na biblioteca para ajustes nos parâmetros indicados, de modo a tentar retornar ao *conteúdo de dados* do arquivo original.

Os dados geoespaciais *raster* foram convertidos pela biblioteca GDAL/OGR por meio dos utilitários **gdal_translate**, **gdalwarp** e **gdalbuildvrt** e comparados quanto à resolução espacial e radiométrica, referência espacial e atributos de elevação, antes e após as conversões.

Posteriormente, a GDAL/OGR foi utilizada para o armazenamento de dados matriciais e vetoriais no formato de banco de dados Spatialite, e sua recuperação gráfica testada no QGIS e Global Mapper. Utilizamos, por fim, a RasterLite e o Spatialite GUI para analisar, comparativamente, a performance da biblioteca com estes aplicativos.

Os formatos escolhidos para análise são representativos de SIG livres e proprietários utilizados no país: TerraView, Spring, QGIS, gvSIG, ArcGIS, ILWIS, MapInfo, Global Mapper, Surfer e Google Earth.

Em todos os testes efetuados, a utilização da GDAL/OGR na linha de comando (de forma independente aos SIG empregados), foi mais eficiente que sua implementação gráfica nas diferentes versões do QGIS analisadas. A linha de comando supera também a implementação gráfica da biblioteca no QGIS em número de recursos de conversão e processamento de dados geográficos.

No primeiro capítulo é apresentada a fundamentação teórico-metodológica da pesquisa, em que são trabalhados os conceitos relacionados à interoperabilidade, bem como apresentada a arquitetura básica da biblioteca GDAL/OGR.

No segundo capítulo, são descritas as estruturas geométricas e topológicas empregadas pela GDAL/OGR na representação digital da realidade geográfica.

No terceiro capítulo, são apresentados as estruturas internas dos formatos de dados escolhidos para análise.

No quarto capítulo são realizados e analisados os processos de conversão sintática envolvendo os dados controlados, e efetuados os processamentos para correção dos arquivos pós-conversão. Neste capítulo apresentamos a metodologia para a superação de problemas de integração de dados geoespaciais em SIG, envolvendo os diferentes formatos testados.

No quinto capítulo é avaliada a integração da GDAL/OGR com o Spatialite (sistema gerenciador de banco de dados espaciais) e, finalmente, no último capítulo são feitas as conclusões finais relacionadas à tese.

Materiais e técnicas empregadas

Neste trabalho, são utilizados os seguintes sistemas e dados:

a) Sistemas:

a.1) Ubuntu Linux 64-bit, versão 13.04, Kernel 3.8.8, ambiente gráfico KDE 4.10.5;

a.2) Bibliotecas GDAL/OGR versão 1.10.0 de 24/04/2013 e 1.11.1, compiladas para os sistemas operacionais Ubuntu Linux 64-Bit e Windows 7 64-Bit;

a.3) QGIS: Versão 1.8.0-Lisboa, compilado sobre GDAL/OGR 1.10.0; QGIS versão 2.0.1-Dufour, compilado sobre GDAL/OGR 1.10.0; QGIS versão 2.8.0-Wien-Desenvolvimento e QGIS versão 2.8.1-Wien, compilados sobre a GDAL/OGR 1.11.1;

a.4) OpenJUMP: versão 1.6.3, distribuição rev.3576 de 29/05/2013;

a.5) DraftSight: versão V1R3.2 de 32-Bit, rodando sobre bibliotecas 32-Bit no Ubuntu Linux 64-Bit;

a.6) GRASS GIS 6.4.3, versão 64-Bit;

a.7) Global Mapper, V16.1, 32-Bit;

a.8) RasterLite 2.3;

a.9) SpatiaLite, versões 4.1.0 e 4.2.0. SpatiaLite GUI versão 4.2.1, e;

a.10) Google Earth, versão 7.1.2.2041.

b) Dados geoespaciais:

b.1) Formato DXF: Mapa Digital da Cidade (MDC) de 2005, arquivo 3313-164. A base está restituída na escala de 1:1000, referenciada aos DATUM horizontal e

vertical SAD-69 e Imbituba, respectivamente, na projeção UTM e possui camadas com curvas de nível, pontos cotados, hidrografia, edifícios, sistema viário, entre outros (total de 97 layers), todos condensados num único arquivo CAD. A versão do formato e em quais sistemas CAD os arquivos do MDC foram gerados, são desconhecidos. Tamanho em disco: 3,5 MB. Fonte: Prefeitura do município de São Paulo.

b.2) Formato ESRI Shapefile: 1 *layer* correspondente ao Setor 082 do MDC 1:1.000 de 2005, referenciado aos DATUM SAD-69 e Imbituba, na projeção UTM. O *layer* contém informações como área projetada e altura dos edifícios. Este setor foi separado dos demais na base municipal, por meio de operação em SIG (recurso de divisão de camada vetorial no QGIS). Tamanho em disco: 3,8 MB. Fonte: Prefeitura do município de São Paulo.

b.3) Formato ArcInfo Grid: Modelo Numérico de Terreno (MNT) da Região Metropolitana de São Paulo (RMSP), com resolução vertical de 5m para a área conurbada, compreendendo uma área de 2500km² e 10m para as áreas periféricas, com área de 5500km² e compatíveis com as precisões altimétricas das cartas 1:10.000 e 1:25.000. Fonte: Instituto Geológico, Secretaria de Meio Ambiente do Estado de São Paulo.

b.4) Formato TIFF: Ortofotos da EMPLASA (Empresa Metropolitana de Planejamento Metropolitano), escala 1:5000, abrangendo a Região Metropolitana de São Paulo, ano de 2007.

CAPÍTULO 1 - FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA

1.1 Representação computacional de dados geográficos

A realidade geográfica é complexa, tendo em vista a quantidade diversificada de fenômenos naturais e antropogênicos de implicação espacial, com sua variação constante no tempo e em escala. A quantidade de detalhes capazes de serem conhecidos e representados, portanto, pode ser ilimitada.

Sistemas computacionais, ao contrário, são fisicamente limitados em sua estrutura de armazenamento e manipulação de dados, implicando que considerações sobre a realidade geográfica devem, teoricamente, restringir a quantidade dos detalhes capturados para sua representação digital (nas estruturas matriciais e vetoriais).

Da concepção de simplificação da realidade geográfica para seu entendimento e representação digital, nascem as visões de campos (contínuos) e objetos (discretos) que, por fim, são reduzidas pelos SIG a uma combinação de 0 e 1, ou seja, codificadas como dados binários nos formatos de arquivos. (Longley et al, 2005).

Basicamente, três são os aspectos gerais envolvidos na representação computacional da realidade geográfica:

- 1) conhecimento dos fenômenos da realidade geográfica (por usuários e/ou desenvolvedores);
- 2) criação de modelos de representação digital destes fenômenos, e;
- 3) estruturação de formas próprias (formatos) de armazenamento destes modelos, nos distintos SIG.

A figura 1 ilustra esses três aspectos fundamentais:

Figura 1. Estágios na representação de dados geográficos em ambiente computacional. Adaptado de Burrough (1989).

Como vimos, a percepção da realidade geográfica influencia sua representação computacional que, por fim, condiciona a própria estrutura de armazenamento dos dados digitais relacionados. Burrough (1989, op.cit.) afirma que os dados geográficos, ao contrário daqueles manipulados em outros sistemas de informação, incluem informações sobre as geometrias empregadas, possíveis conexões topológicas, posição na superfície terrestre e atributos dos objetos representados.

O Decreto 6.666/08 que institui a Infraestrutura Nacional de Dados Espaciais (INDE) no Brasil, amplia o entendimento do dado geoespacial como:

“[...] aquele que se distingue essencialmente pela componente espacial, que associa a cada entidade ou fenômeno uma localização na Terra, traduzida por sistema geodésico de referência⁴, em dado instantâneo ou período de tempo, podendo ser derivado, entre outras fontes, das tecnologias de levantamento, inclusive as associadas a sistemas globais de posicionamento apoiados por satélites, bem como de mapeamento ou de sensoriamento remoto.”.

⁴ Os termos ‘sistema geodésico de referência’, ‘sistema de referência de coordenadas (SRC)’, ‘referência geodésica’, ‘(sistema de) referência espacial’ e ‘*spatial reference system* (SRS)’ são sinônimos.

enquanto no MapInfo TAB binário, não existem referências explícitas sobre os dados no arquivo.

Essa estrutura resulta em arquivos distintos, com extensões diferenciadas em disco (magnético, ótico, etc).

Na segunda, de *alto nível* (do ponto de vista computacional, pois está mais próxima ao usuário) a realidade geográfica é separada em camadas que representam um tema, um aspecto desta realidade. As diferentes camadas georreferenciadas formam o *mapa* digital em SIG, como ilustrado na figura 3:

Figura 3. Estrutura de *layers* (camadas) dos dados geoespaciais. Adaptado de *ESRI Web Help*.

Essencialmente, as manipulações e modificações nos *layers* em SIG afetam diretamente os arquivos correspondentes em disco, em seus formatos de dados respectivos.

1.2 Integração de dados geoespaciais em SIG

A questão da integração (ou reaproveitamento) de dados geoespaciais existentes é central e recorrente na prática profissional e de pesquisa em Geografia, Cartografia e Geoprocessamento. Em diversos estudos e aplicações, é

absolutamente necessária a incorporação de mapas vetoriais (representando hidrografia, curvas de nível, lotes e edificações, por exemplo), assim como imagens de satélites, ortofotos, MNT, entre outros produtos cartográficos, para seu início ou pleno desenvolvimento. Seria impraticável ter que iniciar diferentes levantamentos para cada pesquisa a ser desenvolvida.

Nessa linha, a CONCAR estabelece:

“[...] Com a multiplicidade de geotecnologias existentes no mercado, a produção de dados geoespaciais e sua distribuição tornam-se mais ágeis a cada dia. No entanto, para isso os dados necessitam ser gerados segundo padrões e especificações técnicas que garantam o compartilhamento, a interoperabilidade e a disseminação destes dados, sendo estes aspectos fundamentais em uma Infra-estrutura de Dados Espaciais (IDE)” (INDE, Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais (ET-EDGV), versão 2.1.3, 2008).

Desenvolvendo a ideia anterior com mais detalhes, a INDE coloca:

“A complexidade da produção e disseminação, inerente à própria natureza do dado geoespacial, tem trazido dificuldades para instituições e pesquisadores interessados no reaproveitamento de dados já trabalhados em outros projetos, planos, produtos ou programas. **Alguns fatores dificultam a reutilização dos dados.**” (Plano de Ação da INDE, p. 69, Capítulo 4. O destaque é nosso).

Ou seja, a dificuldade em reutilização dos dados resulta em um problema típico de interoperabilidade⁵.

Entre os fatores que dificultam a reutilização ou integração de dados, podemos apontar: diferentes formatos de arquivos; falta de padrões estabelecidos; inconsistências geométricas, topológicas e de referência espacial; significado do dado para a comunidade de usuários, modelos conceituais⁶ distintos, entre outros. (Júnior, 2002)

5 Interoperabilidade significa a capacidade de compartilhar informações entre diferentes sistemas e usuários. (Goodchild et al, 1997).

6 Modelos conceituais são representações abstratas de uma aplicação, independentemente de *hardware* e *software*, tendo em vista facilitar o entendimento entre desenvolvedores e usuários. Mais adiante, voltaremos ao assunto.

Tendo em vista a integração completa de dados geoespaciais, Mohammadi et al. (op.cit., 2010), estabelecem três níveis de interoperabilidade:

- 1) sintático: o nível mais básico, que diz respeito à conversão direta entre formatos de dados;
- 2) estrutural: o nível intermediário, que diz respeito ao mapeamento entre modelos conceituais, e;
- 3) semântico: o nível mais alto, que diz respeito ao *significado* dos dados compartilhados, entre usuários de sistemas heterogêneos.

A interoperabilidade completa seria alcançada pela integração desses níveis, como ilustra a figura 4:

Figura 4. Hierarquia para o alcance da integração ideal de dados. Fonte: Mohammadi et al. (2010).

O nível sintático é o que diretamente mais influencia a interoperabilidade em SIG, dado que grande parte das padronizações desenvolvidas (notadamente pelo OGC⁷) está relacionada aos tipos de formatos, geometrias, referência espacial e

⁷ O *Open Geospatial Consortium* (OGC) é um consórcio internacional que agrupa empresas, instituições governamentais e acadêmicas com o objetivo de promover interoperabilidade entre sistemas envolvendo dados geoespaciais. O OGC define diversas

topologias suportadas. A biblioteca GDAL/OGR atua nestes aspectos. Adiante, veremos as definições do OGC para os elementos citados.

É usual que, devido a heterogeneidade dos formatos e a performance dos aplicativos, as conversões possam não ser plenamente eficientes, gerando problemas nos arquivos resultantes, sendo necessária então a utilização de recursos para a correção dos erros apresentados. A figura 5 ilustra o processo usualmente aplicado para as conversões sintáticas e correção dos erros.

Figura 5. Etapas para a integração de dados. Fonte: Mohammadi et al. (2010)

No nível estrutural, os modelos conceituais realizam o processo de abstração da realidade geográfica, de tal forma que somente seus elementos essenciais sejam enfatizados, enquanto os não essenciais sejam descartados. Ou seja, eles estabelecem *quais* e *como* devem ser representados os fenômenos geográficos. O emprego de diferentes modelos⁸ em aplicações SIG representa, na prática, formas

especificações, que podem ser divididas em duas categorias: (a) especificação abstrata, que define o modelo de geometrias e serviços desconsiderando detalhes de sua implementação, e; (b) especificação de implementação, que mostra como os conceitos da especificação abstrata podem ser implementados utilizando tecnologias de desenvolvimento existentes. (CARNEIRO et al, 2008).

diferenciadas de conceber a realidade geográfica e sua consequente representação computacional.

O nível semântico, por sua vez, diz respeito ao significado (humano) de categorias empregadas na descrição de dados geográficos. Seu desenvolvimento acompanha o processo de consolidação de Infraestruturas de Dados Espaciais (IDE) em diversos países, com o objetivo de promover a interoperabilidade e diminuir os custos referentes a aquisição e geração (redundantes) de dados. No país, os esforços da INDE estão direcionados, sobretudo, em padronizar essas categorias por meios dos perfis MGB (Metadados Geoespaciais do Brasil).

Na prática, a questão semântica é colocada da seguinte maneira: o que é a categoria *x*? Como descrevê-la? Quais temas podem ser agrupados a ela?

O exemplo, abaixo, mostra a ampliação do entendimento sobre a categoria ‘Limites Administrativos’ por meio do agrupamento da categoria 033 à 003, definidas no manual de preenchimento dos perfis sumarizados.

a) Limites Administrativos:

- **Entendimento *inicial*:** limitesAdministrativos (003) - limites legais do território. Exemplos: fronteiras administrativas e políticas;

- **Entendimento *ampliado*:** limitesPoliticoAdministrativos (Nacional, Estaduais, Municipais e Distritais) – **(033)** – composto por: Limites Nacional / Relações Exteriores, Limites Regionais, Limites Estaduais, Limites Municipais, Limites Distritais, Perímetro urbano, Limites operacionais e Limites naturais;

Este tema interessa a várias comunidades de usuários (prefeituras, governos estaduais, IBGE, ministérios, empresas, etc), sendo que cada uma pode atribuir-lhe um determinado significado. A padronização é necessária para evitar que sentidos divergentes ou mesmo contraditórios sejam-lhes atribuídos, ocasionando problemas na reutilização dos dados.

8 O *Simple Features Specification* (SFS) do OGC (especificação abstrata), que veremos adiante; e o OMT-G (Borges, 1997), ambos baseados nas visões de *campos* e *objetos*, são exemplos de modelos conceituais em SIG.

Percebemos, assim, que o alcance da interoperabilidade plena, em todos os seus níveis integrantes, não é uma tarefa simples e exige um conjunto de fatores relacionados, que vão desde a estruturação dos formatos de dados utilizados, até a padronização das categorias que irão descrevê-los.

Recentemente (entre novembro de 2014 e janeiro de 2015), a revista internacional *Geospatial World*⁹ conduziu uma pesquisa com seiscentos e cinquenta leitores sobre o uso de *softwares* de SIG, GNSS, Sensoriamento Remoto, 3D, entre outros. Para a questão que versava sobre *quais seriam os fatores mais determinantes* para a aquisição dos aplicativos, a resposta obtida entre 84% dos entrevistados foi o fator “interoperabilidade”, superando outros como preço, suporte técnico, fácil disponibilidade e escolha.

Este é, portanto, um dos aspectos fundamentais no que diz respeito a escolha de *softwares* na área do Geoprocessamento atualmente.

Para o contexto desta pesquisa, nossos testes estarão direcionados ao aspecto sintático da interoperabilidade, dado que estudaremos em que medida a biblioteca GDAL/OGR pode permitir uma integração efetiva de dados em ambiente SIG, por meio de comparações nas *feições*¹⁰ e nas *matrizes*, antes e após as conversões.

A seguir, será apresentada em detalhes a estrutura da GDAL/OGR.

1.3 O FOSS (Free and Open Source Software) e a GDAL/OGR

O FOSS (*Free and Open Source Software*)¹¹, mais conhecido no Brasil como “*software* livre”, é uma realidade consolidada em diversas áreas da informática: na área de sistemas operacionais, editores de texto, processadores de imagens, sistemas gerenciadores de banco de dados (SGBD), desenhos vetoriais, etc.

9 Geospatial World Annual Readers’ Survey 2014. Geospatial World é um dos maiores portais técnicos da área geoespacial.

10 Feições (geográficas ou geoespaciais) são objetos vetoriais do tipo Point, LineString e Polygon (Longley et al. 2005). Mais detalhes sobre essas geometrias, serão apresentados posteriormente.

11 A expressão FOSS4G (*Free and Open Source Software for Geospatial*) é também empregada, indicando o uso do FOSS no campo do Geoprocessamento.

Entre as vantagens de utilização de sistemas livres estão:

- 1) utilização para quaisquer finalidades: educacional, comercial, científica, etc;
- 2) adaptação dos códigos às necessidades do usuário;
- 3) distribuição gratuita de cópias, e;
- 4) possibilidade de melhoria do *software*, disponibilizando-o ao público interessado.

Figura 6. Área de interação entre as licenças de *software* “livres”, “*open*” e proprietárias. Fonte: Stefan Steiniger e Erwan Bocher (2009).

Basicamente, são as licenças que controlam como podem ser usados os sistemas e/ou terem seu código alterado. Enquanto os sistemas “proprietários” (*proprietary software*) possuem código fonte fechado e ao usuário é garantido somente o “uso da licença” e não sua propriedade, os sistemas “*open source*” (disponibilizados gratuitamente) possuem licenças mais flexíveis e semelhantes entre si.

As licenças GPL (*GNU General Public License*) e LGPL (*GNU Lesser General Public License*), por exemplo, permitem a alteração no código fonte do sistema, desde que estas modificações possam ser reaproveitadas no código e disponibilizadas gratuitamente (por isso, impondo restrições a sua incorporação em sistemas proprietários); a licença BSD (*Berkeley Software Distribution*) permite alterações no código porém estas não precisam ser, obrigatoriamente, disponibilizadas no código fonte, e; a licença MIT (*Massachusetts Institute of*

Technology) permite, sem restrição, a utilização do código tanto em sistemas livres quanto proprietários.

O campo da Tecnologia da Informação Geográfica é um dos grandes beneficiados com o FOSS, possuindo sistemas operando sobre plataformas distintas (distribuições Linux, Windows, OS X, etc), com milhares de usuário e contribuidores em todo o mundo.

Neste trabalho utilizaremos alguns dos principais FOSS disponíveis na área do Geoprocessamento na atualidade, como o QGIS, OpenJUMP e GRASS.

Em seguida, será descrita em detalhes a estrutura da biblioteca GDAL/OGR, também sob a orientação da FOSS (licença MIT).

1.4 Biblioteca GDAL/OGR

Geospatial Data Abstraction Library e *Simple Features Library* (GDAL/OGR) é uma biblioteca utilizada para a conversão sintática e processamento de dados geoespaciais (matriciais e vetoriais). Seus recursos podem ser adicionados por meio de programação¹² a diferentes *softwares* (livres e proprietários); adicionalmente, podem ser compiladas e utilizadas via linha de comando em sistemas operacionais específicos (são suportados os principais sistemas: Linux, OS X e Windows).

A biblioteca GDAL/OGR representa, atualmente, o projeto FOSS4G de maior alcance na questão da interoperabilidade geoespacial, sendo utilizada por sistemas livres mundialmente conhecidos como o SAGA GIS, QGIS, GRASS, GeoServer e MapServer. Seus recursos estão também implementados em sistemas proprietários como o FME e ArcGIS¹³, por exemplo.

A utilização da biblioteca *compilada* nesta pesquisa significa que os recursos de conversão e processamento de dados geoespaciais, estarão disponíveis por linha de comando no terminal do sistema operacional, sendo operados de maneira *autônoma* em relação aos SIG utilizados em nossos testes. Com isto, poderemos

12 Entre as linguagens disponíveis estão: C++, C#, JAVA e Python.

13 Não sabemos se os recursos implementados foram modificados em relação ao código original.

utilizar funções adicionais da biblioteca que, eventualmente, não tenham sido implementadas por códigos de programação nos sistemas testados neste trabalho.

É importante mencionar que a GDAL/OGR é configurada automaticamente quando ocorre a instalação do QGIS nos sistemas operacionais suportados, cabendo ao usuário apenas operá-la pela via de comando nos terminais (linha de comando) dos respectivos sistemas.

No Windows, existe o aplicativo OSGeo4W Shell que acessa os recursos da biblioteca; no Linux, os recursos da GDAL/OGR são acessadas pelo terminal. A figura 7 ilustra o exposto:

```

OSGeo4W Shell
--( OSGeo4W Shell Commands )--

assistant          ogrtindex
avcelexport         osgeo4w-setup
avcinport           pal2rgb
bmp2tiff            pixeltool
cs2cs               ppm2tiff
curl                proj
designer             psq1
dllupdate           pylupdate4
example1            pygcc4
example2            python
fax2ps              pythonw
fax2tiff            qcollectiongenerator
fftw-wisdom         qdoc3
gdaladd             qgis-bin
gdalbuildvrt        qgis-browser-bin
gdaldem             qhelpconverter
gdalenhance         qhelpgenerator
gdalinfo            qmake
gdallocationinfo   qmlplugindump
gdalmanage          qmlviewer
gdalserver          qt3to4
gdalsrsinfo         qttracereplay
gdalindex           ras2tiff
gdaltransform       ras2tiff
gdalwarp            rce
gdal_contour        rgb2ycbcr
gdal_grid           rpcgen
gdal_rasterize      sip
gdal_translate      spatialite_gui_4.2.1
geod                testpsg
geotifcp            textreplace
gif2tiff            thumbnail
gltpd              tiff2bw
gpsbabel            tiff2pdf
gwin32c             tiff2ps
idc                 tiff2rgba
lconvert            tiffcmp
linguist            tiffcp
listgeo             tiffcrop
lrelease            tiffdither
lupdate             tiffdump
moc                 tiffinfo
nad2bin             tiffmedian
nccopy              tiffset
ncdump              tiffsplit
ncgen               uic
ncgen3              uic3
nearblack           vcredist-2013-x64
nircmd              xmlcatalog
nircmdc             xllint
ogdi_import         xmlpatterns
ogdi_info           xmlpatternsvalidator
ogr2ogr             xmlwf
ogrinfo             xxmmlink
ogrlineref

epsq_tr             gdal_fillnodata   ps2pdfi2
esri2vrt            gdal_merge         ps2pdfi3
gcp2vuc             gdal_polygonize    ps2pdfi4
gcp2vuld            gdal_proximity     ps2pdfxx
gdal2tiles           gdal_retile         pyuic4
gdal2xyz             gdal_sieve          qgis-browser
gdalchksum           grass64             qgis
gdalcompare          gsetgs              rgb2pct
gdalident            make-bat-for-py     saga_gui
gdalimport           mkgraticule         setup-test
gdalmove             o-help              setup
gdal_auth            o4w_env
gdal_calc            pct2rgb
gdal_edit            ps2pdf

```

GDAL 1.11.1, released 2014/09/24


```

OGR VRT: bash - Konsole
Arquivo  Editar  Exibir  Favoritos  Configurações  Ajuda
Layer (String) = 626_SEGW_CURVA_INTERM
SubClasses (String) = AcDbEntity:AcDbPolyline
ExtendedEntity (String) = (null)
Linetype (String) = (null)
EntityHandle (String) = 1030
Text (String) = (null)
Style = PEN(c:#000000)
LINESTRING (324205.880857584474143 7393275.799464007839561 726,324205.479999996779952 7393276.05 726,324204.549999997077975 7393276.5899
99997057021 726,324203.78888610569993 7393277.057047161273658 726)

OGRFeature(3313-164_linhas):584
Layer (String) = 604_CURVA_INTERM
SubClasses (String) = AcDbEntity:AcDbPolyline
ExtendedEntity (String) = (null)
Linetype (String) = (null)
EntityHandle (String) = 1031
Text (String) = (null)
Style = PEN(c:#000000)
LINESTRING (324508.267260588181671 7393147.560646720230579 728,324508.317855838686228 7393147.910260180011392 728,324508.035854656191077
7393149.677430837415159 728,324507.825383500778116 7393150.360183217562735 728)

OGRFeature(3313-164_linhas):585
Layer (String) = 626_SEGW_CURVA_INTERM
SubClasses (String) = AcDbEntity:AcDbPolyline
ExtendedEntity (String) = (null)
Linetype (String) = (null)
EntityHandle (String) = 1032
Text (String) = (null)
Style = PEN(c:#000000)
LINESTRING (324507.825383500778116 7393150.360183217562735 728,324507.799598803278059 7393150.443826812319458 728,324506.779678563005291
7393152.583956732414663 728)
OGR VRT: bash

```

Figura 7. Exemplo da linha de comando no Windows, acima, listando os recursos da GDAL/OGR; abaixo, o terminal no Linux (aplicativo Konsole), indicando as propriedades geométricas, coordenadas espaciais e atributos de feições vetoriais, após a entrada de um comando da biblioteca.

Uma vez compilada para o sistema operacional, os recursos da GDAL/OGR podem ser chamados a qualquer momento pela linha de comando e utilizados conjuntamente ao Linux, Windows e OSX.

Sendo o QGIS o SIG livre mais utilizado atualmente¹⁴, os recursos de integração da GDAL/OGR com o SIG podem, teoricamente, beneficiar os milhares de usuários do sistema, mundialmente.

A partir daqui, para as conversões em dados vetoriais, utilizaremos apenas o termo OGR; para os dados matriciais, o termo GDAL.

A lista para as centenas de formatos matriciais e vetoriais suportados pela biblioteca, pode ser consultada em: <<http://www.gdal.org>>. Acesso em 9 mar. 2015.

1.5.1 Arquitetura da OGR

As classes do modelo vetorial (OGR) são responsáveis pelos recursos de interpretação geométrica, de transformação entre diferentes referenciais geodésicos

¹⁴ Segundo pesquisa recente publicada pelo GEOeduc e coordenada pelo Dr. Philip Davis, gerente de projetos do consórcio *National Information, Security & Geospatial Technologies*.

e suporte a diversos formatos de arquivo na biblioteca. A OGR é estruturada conforme segue:

a) Classe de Geometria (OGRGeometry): oferece suporte a elementos geométricos, utilizando a *Simple Features Especification* (SFS) do OGC como *Point*, *Multipoint*, *LineString*, *MultiLineString*, *Polygon*, *MultiPolygon*, *GeometryCollection* e *Surface*. Essas geometrias serão apresentadas em detalhes posteriormente.

b) Classe de Referência Espacial (OGRSpatialReference): implementa a definição do OGC¹⁵ e estabelece os parâmetros para acesso e transformações entre referenciais geodésicos¹⁶, em que são suportadas coordenadas geográficas, projetadas, locais, verticais e geocêntricas.

c) Classe de Feição (OGRFeature): enquanto a OGRGeometry reconhece a geometria de uma feição vetorial e sua posição espacial, a OGRFeature interpreta a geometria e lhe adiciona atributos e um identificador único de feição (Feature ID), para distingui-la das demais no *layer*. Os tipos e nomes dos atributos são referenciados a uma classe derivada, a OGRFeatureDefn.

d) Classe de *Layer* (OGRLayerClass): representa um *layer* na base de dados. Todas as feições nesta classe possuem a mesma definição (dada pela OGRFeatureDefn). Basicamente, esta classe contém métodos para a leitura de feições nos formatos de arquivos.

e) Classe de Base de Dados (OGRDataSource): faz referência a uma série de objetos OGRLayer. Na prática, esta classe faz referência a um único arquivo, conjunto de arquivos ou bancos de dados.

¹⁵ Esta definição do OGC fornece uma documentação detalhada para implementar em *software*, recursos de acesso e transformação entre referenciais geodésicos em dados geoespaciais.

¹⁶ Estes recursos são desenvolvidos sobre a biblioteca PROJ.4 mantida pela OSGEO (*Open Source Geospatial Foundation*).

f) Classe de Drivers (OGRSFDrivers): é aplicada para cada formato de dado a ser suportado, juntamente com as classes OGRDataSource e OGRLayer.

As classes descritas são implementadas nos códigos de programação para a integração da OGR em SIG (por exemplo, no menu ‘Salvar camada vetorial...’ no QGIS).

Os aplicativos **ogr2ogr** e **ogrinfo**, por sua vez, também incorporam aquelas definições, o que significa que os recursos da OGR se tornam disponíveis pela linha de comando no terminal Linux, isto é, são aplicativos *executáveis*. A lista abaixo enumera algumas das possibilidades para os aplicativos:

- **ogr2ogr -f “nome_do_formato” formato_de_saída.extensão**
formato_de_entrada.extensão: utilizado para a conversão entre distintos formatos de dados. Alguns valores possíveis são:
 - f "ESRI Shapefile"
 - f "MapInfo File"
 - f "GML"
 - f "KML"
- **ogr2ogr -t_srs srs_def**: utilizado para transformações entre referenciais geodésicos, utilizando o código EPSG¹⁷.
- **ogr2ogr -zfield nome_do_campo**: utiliza o campo especificado para adicionar valores de altitude (z) às geometrias.

¹⁷ *European Petroleum Survey Group* (EPSG) é uma organização científica europeia que compilou e disseminou o *EPSG Geodetic Parameter*, uma ampla base de dados utilizadas como padrão de interoperabilidade, ao atribuir códigos únicos (nunca repetidos) às referências geodésicas utilizadas nos diversos países.

- **ogrinfo -so**: lista informações sumarizadas sobre arquivos em formatos de dados suportados.
- **ogrinfo -al**: lista informações detalhadas sobre arquivos em formatos de dados suportados.

Para a lista completa de recursos, consultar o endereço: <<http://www.gdal.org>>.

Acesso em 9 mar. 2015.

Outro recurso interessante da biblioteca é a estruturação de formatos vetoriais genéricos chamados de **OGR VRT** (*Virtual Format*). Por meio dele, é possível criar códigos XML¹⁸, gerar arquivos “virtuais” no formato OGR VRT capazes de reconhecer os demais *drivers* suportados pela biblioteca (ESRI Shapefile, MIF/MID, DXF, GML, KML, DGN, etc), configurá-los de acordo com as regras descritas no código e, por fim, carregá-los no QGIS¹⁹.

O OGR VRT funciona, fundamentalmente, como um formato vetorial genérico, capaz de interpretar e extrair informações espaciais (geométricas, de atributos, de referência espacial, etc) de diversos outros formatos, utilizando a arquitetura básica da biblioteca.

18 XML (*Extensible Markup Language*) é uma linguagem de marcação recomendada pela W3C para a criação de documentos com dados organizados hierarquicamente, tais como textos, banco de dados ou desenhos vetoriais. Para maiores detalhes:

<<http://www.w3.org/>> Acesso em 12 mar. 2015. A Safe Software Inc (empresa desenvolvedora do FME) lista sete motivos para a adoção do XML como formato de dados geoespaciais: 1) padrão aberto; 2) humanamente inteligível; 3) fácil validação; 4) pode representar qualquer tipo de dado; 5) é a linguagem padrão de serviços *Web* e de comunicação entre sensores; 6) abundância de bibliotecas para leitura / escrita no formato, e; 7) suporte nativo para caracteres internacionais. Mais detalhes em: <<http://blog.safe.com/2014/07/reasons-to-love-xml-json/>>. Acesso em 12 mar. 2015.

19 O QGIS, sendo desenvolvido sobre a biblioteca, pode ser utilizado como o visualizador de mapas, atributos e metadados do formato OGR VRT.

1.5.2 Arquitetura da GDAL

As classes do modelo matricial (GDAL) da biblioteca estão ligadas aos recursos de interpretação, conversão de formatos e processamento de dados matriciais. Da mesma maneira que a OGR, a implementação desses recursos depende da chamada das classes relacionadas nos códigos de programação, tendo em vista integrá-los em SIG (como os menus 'Raster' e 'Salvar como...' no QGIS) ou utilizá-los por meio dos utilitários. Diferentemente da OGR, existem 28 utilitários para a aplicação dos recursos da GDAL na linha de comando, entre eles:

- **gdalinfo** (reporta informações sobre arquivos);
- **gdal_translate** (converte arquivos matriciais para outros formatos de dados especificados);
- **gdalwarp** (realiza transformações entre referenciais geodésicos);
- **gdal_contour** (extrai isolinhas a partir de Modelos Numéricos de Terreno), entre outros recursos.

Para a lista completa, consultar o endereço: <<http://www.gdal.org>>. Acesso em 9 mar. 2015.

A principal classe da GDAL é estruturada conforme segue:

a) **Classe de Base de Dados** (GDALDataSet): diz respeito a um conjunto de bandas *Raster* e suas informações comuns, como o tamanho da matriz (em *pixel* e linhas) e especificação do referencial geodésico (Coordinate System) para a conversão do georreferenciamento. A classe suporta a utilização de metadados.

A implementação da Coordinate System na GDAL, tal como a OGRSpatialReference na OGR, segue a especificação da OCG Coordinate Transformation Service e possui duas maneiras de descrever o relacionamento entre posições dos *pixels* nas bandas:

- a) em coordenadas de *pixel*/linha e;
- b) em coordenadas espaciais (georreferenciadas).

Na GDAL, a função mais utilizada para transformações entre sistemas geodésicos de referência distintos é a transformação afim, que relaciona coordenadas dos *pixels* a coordenadas espaciais (georreferenciadas).

Esta transformação tem o efeito de deslocar (em x,y), rotacionar (de 0 a 360 graus) e escalonar bases matriciais com a inserção de, no mínimo, seis coeficientes (equações de x' e y'), como ilustra a figura 8:

Figura 8. Esquema conceitual de transformação polinomial de primeiro grau (afim), realizada pela GDAL para estabelecer o geoposicionamento das bandas matriciais. Fonte: Adaptado de ESRI WebHelp.

A transformação é realizada não somente pelo utilitário **gdalwarp** mas toda vez que é preciso gerar um novo arquivo (seja pelo utilitário **gdal_translate**, seja pelo recurso de ‘Salvar Raster como...’ no QGIS), uma vez que para a conversão entre formatos, a biblioteca precisa interpretar o arquivo original e reescrevê-lo, mapeando suas coordenadas originais para o formato de destino. Esta operação resultará em novos arquivos em disco.

É possível, assim como no OGR VRT, estabelecer e configurar o formato virtual para bases matriciais (**GDAL VRT**).

Em seguida, serão apresentadas as estruturas geométricas suportadas pela GDAL/OGR, baseadas na especificação abstrata do OGC.

CAPÍTULO 2 - DEFINIÇÕES GEOMÉTRICAS

2.1 Geometrias suportadas pelo *Simple Features Specification*

Dada a diversidade de geometrias utilizadas (tanto as simples como pontos, retas e polígonos, quanto as mais complexas como arcos, círculos, elipses e sólidos 3D [Davis, 2001]), modelos conceituais e sistemas existentes no campo do Geoprocessamento, a padronização se fez necessária, fazendo surgir no âmbito do OGC a especificação abstrata *Simple Features Specification* (SFS), que resulta na definição rígida das entidades geométricas e de seus relacionamentos topológicos, independentemente de plataforma SIG ou formato de dados. Neste modelo conceitual, cada objeto geométrico está associado a um sistema de referência espacial, o qual descreve o espaço de coordenadas em que aquele está inserido, como mostra a figura 9:

Figura 9. Modelo conceitual Simple Features do OGC. Notar os relacionamentos entre as geometrias empregadas. Fonte: OGC Simple Features.

O conhecimento das definições geométricas do *Simple Features* é fundamental para o esquema de conversão sintática entre formatos de dados geoespaciais analisados neste trabalho, uma vez que a biblioteca GDAL/OGR faz uso deste modelo conceitual.

As geometrias suportadas pelo SFS de interesse da pesquisa serão descritas a seguir, mantendo-se os nomes inalterados, uma vez que a nomenclatura é preservada na estruturação dos formatos binários e de linguagens de marcação:

a) Point e MultiPoint

Point é um objeto geométrico de dimensão zero e representa uma única localização no espaço de coordenadas. Num sistema geodésico de referência, pode receber valores de z (altitude) e m (medida para a operação de referenciamento linear, interpolação).

MultiPoint é uma coleção de Points.

Figura 10. Exemplo de Point e MultiPoint, à esquerda e direita, respectivamente.

b) Lines, LineString, MultiLineString e LinearRing

b.1) Line: é um objeto unidimensional formado por exatamente dois pontos (segmento de reta).

b.2) LineString: é formada por ajustes de seguidas Lines entre os pontos iniciais e finais da geometria. É também conhecida como *polyline* (poli-linha). Cada par consecutivo de pontos, define um segmento de reta (Line).

Figura 11. Ilustração de uma LineString, formada por diversas Lines. Fonte: adaptado de GRASS Wiki.

b.3) MultiLineString: A coleção de polylines forma a MultiLineString.

b.4) LinearRing: é uma polyline fechada em que suas coordenadas iniciais e finais são as mesmas. É empregado como o limite exterior ou interior de um polígono (no último caso, formando o polígono com buracos).

Figura 12. Exemplos de LinearRing, com apenas os vértices iniciais e finais. Fonte: Adaptado de OGC Simple Features versão 1.2.1.

c) Polygon e MultiPolygon: polígono é um objeto bidimensional definido por um limite externo e 0 ou mais limites internos (buracos). Esses limites internos/externos são formados por LinearRings.

Figura 13. Exemplo de polígono *com* e *sem* buracos. Fonte: OGC Simple Features versão 1.2.1.

d) GeometryCollection

Tipo geométrico formado por coleção de geometrias básicas, como ilustra a figura 14.

Figura 14. Exemplo da coleção de geometrias com as primitivas pontos, linhas e polígonos.

2.2 Geometrias não suportadas pelo *Simple Features Especification*

As geometrias a seguir, apesar de não suportadas pelo SFS do OGC, são componentes de diversos formatos geoespaciais analisados neste trabalho, por isso serão apresentadas:

a) Arc

É uma curva que compõe um círculo.

Figura 15. ACB e ADB são dois arcos. Fonte: iCoachMath.com

b) Elipsis

É formada quando um cone é cortado por um plano.

Figura 16. Exemplo de uma elipse. Notar que, diferentemente de um círculo, o eixo x é maior que o y. Fonte: iCoachMath.com

c) Circle

Círculo é o lugar de todos os pontos que estão a uma mesma distância (equidistantes) de um dado ponto.

Figura 17. O é o centro do círculo. Todos os pontos no círculo como A, B, C, e D, por exemplo, são equidistantes de O, seu centro. Fonte: iCoachMath.com

d) Sólidos geométricos

Como estabelece Fernando Guesser (2011):

“Os sólidos geométricos têm três dimensões: comprimento, largura e altura. [...] são separados do resto do espaço por superfícies que os limitam. E essas superfícies podem ser planas ou curvas. Dentre os sólidos geométricos limitados por superfícies planas, existem os prismas, cubo e as pirâmides. Dentre os sólidos geométricos limitados por superfícies curvas, estão o cilindro, o cone e a esfera, que são também chamados de sólidos de revolução.” (Aula 2, Figuras Geométricas).

Para o contexto deste trabalho, interessa-nos sobretudo o “prisma”, sólido geométrico limitado por polígonos e que pode ser imaginado como o resultado do deslocamento destes (Guesser, 2011, op. cit.), conforme ilustra a figura 18:

Figura 18. Sólidos geométricos. Fonte: Guessier, 2011.

Os sólidos geométricos são muito utilizados em ambiente CAD e, mais recentemente, na modelagem 3D em SIG.

e) Raster (pixel e cell)

Uma base de dados *raster* ou matricial, consiste em uma matriz de células ou *pixels* organizadas em linhas e colunas (gride) em que cada célula, além das coordenadas espaciais, contém valores representando um tema ou a distribuição de valores como altitudes e temperaturas, por exemplo. A figura 19 ilustra o exposto:

Figura 19. Exemplo da estrutura matricial em que a forma de um objeto geográfico é formada por seu conjunto de células. Fonte: ESRI WebHelp.

Ortofotos, MNT, mapas digitalizados (em *scanners*) e imagens de satélites, são exemplos de dados geoespaciais estruturados como *pixels* ou células.

2.3 Relacionamentos topológicos entre feições geográficas

A capacidade de feições geográficas permanecerem inalteradas sob deformação, está relacionada a um tipo de propriedade espacial em SIG conhecida como relação topológica. Quando mapas digitais sofrem mudanças em suas formas geométricas, basicamente sendo esticados ou achatados por operações de reprojeção²⁰, distâncias e ângulos podem sofrer alteração, mas propriedades espaciais como disjunção, contenção ou adjacência permanecem inalteradas.

O seu conhecimento é de fundamental importância ao processo de criação e conversão de formatos vetoriais em SIG, uma vez que delas depende a qualidade cartográfica das bases digitais.

Os problemas topológicos mais comuns em vetores, são:

- Em feições tipo Polygon: geometrias não fechadas (*unclosed*); buracos (*gaps*) entre seus limites comuns e sobreposição (*overlap*). Em SIG que implementam as especificações do OGC, o problema de polígonos abertos não existe pois o sistema recusa a criação da geometria;
- Em feições tipo LineString: não conexão entre seus nós. Se existe um '*gap*' entre eles, o erro é conhecido como '*undershoot*'; se a polyline atravessa a outra e termina além do nó ao qual deveria se conectar, o problema é conhecido como '*overshoot*', e;

²⁰ Nome proveniente do inglês, *reprojection*, que indica as transformações entre sistemas geodésicos de referência distintos, ou seja, operações envolvendo DATUM e projeções cartográficas.

- Em feições tipo LinearRing: geometrias não fechadas (*unclosed*), fazendo com que se transformem em LineStrings; buracos (*gaps*) entre seus limites comuns e sobreposição (*overlap*).

A figura 20, ilustra o exposto:

Figura 20. Erros topológicos mais comuns envolvendo LineStrings (1:undershoot; 2: overshoot) e Polygon (3: gap). Fonte: QGIS.

O armazenamento dos relacionamentos topológicos entre as geometrias da feição, depende da existência de suporte na estrutura do formato, sendo que poucos incluem estes tipos de informações, como é o caso do GML que veremos posteriormente. A figura 21 indica esquematicamente o exposto:

Figura 21. Representação esquemática de dado geoespacial vetorial que inclui informações sobre relacionamentos topológicos entre as feições da camada. Fonte: Longley et al (2005).

O não suporte a topologias pode ser inferido pela de redundância de geometrias entre feições vizinhas, como por exemplo vértices e linhas coincidentes duplicadas ou recobrimento entre polígonos adjacentes. A figura 22 ilustra o processo:

Figura 22. À esquerda, 'overlap' de polígonos com pseudo-topologia; à direita, os mesmos polígonos em formato de dado topológico. No segundo caso, a área de sobreposição é compartilhada por ambos os vetores e não é armazenada de maneira redundante pelo sistema (como acontece no primeiro). Fonte: adaptado de GRASS Wiki.

Em SIG, a forma mais utilizada para evitar erros de topologia em formatos que não a suportam explicitamente, é condicionando o processo de criação de vetores. Este recurso, conhecido como “*planar enforcement*”, garante que todas as geometrias existam *somente* no mesmo plano bidimensional (R2), *forçando* sua quebra em *vértices (nós)* comuns (Theobald, 2001).

O emprego desse recurso resulta, *praticamente*, no compartilhamento de vértices e segmentos de retas entre distintas feições geográficas, como ilustra a figura 23, evitando os erros topológicos mais comuns.

Figura 23. Exemplo de edição vetorial *com* e *sem* o recurso de *Planar Enforcement*. Fonte: adaptado de Understanding Topology and Shapefiles. ArcUser, 2011.

O QGIS estabelece esse processo somente se os recursos de ‘Edição de Topologia’ e ‘Habilitar atração na intersecção’ estiverem ativados na criação de vetores; do contrário, serão geradas geometrias com pseudo-topologia. É possível configurar linhas e polígonos para se aproximarem em seus vértices e segmentos (de reta) compartilhados, garantindo a criação de geometrias sem os problemas apresentados.

No caso de dados originariamente gerados com pseudo-topologia, ou seja, que possam apresentar erros, é preciso empregar um conjunto de recursos existentes em SIG, a fim de corrigí-los nas diferentes feições. Este processo é

conhecido como limpeza topológica (ou *data cleaning*) e está disponível no QGIS (somente os recursos de verificação de problemas) e no OpenJUMP (verificação e correção automática dos erros).

Há alguns SIG como o SPRING, TerraView e GRASS, por exemplo, que implementam o processo de remoção de geometrias duplicadas na importação de dados. Esta é uma tentativa, inicial, de correção de problemas topológicos em bases vetoriais.

O OGC define os relacionamentos topológicos entre objetos com no máximo duas dimensões (dim^{21}), projetados no plano R^2 , da seguinte forma: sejam a e b dois objetos geométricos distintos, $I(a,b)$ representa seus interiores, $B(a,b)$ seus limites e $E(a,b)$ seus exteriores. Assim, temos:

a) *Touch* (Toca): $a.Touch(b) \Leftrightarrow (I(a) \cap I(b) = \emptyset) \wedge (a \cap b) \neq \emptyset$. Os interiores de 'a' e 'b' não se intersectam e a intersecção entre 'a' e 'b' é diferente de nula.

b) *Overlap* (Sobrepõe): $a.Overlaps(b) \Leftrightarrow (dim(I(a)) = dim(I(b)) = dim(I(a) \cap I(b))) \wedge (a \cap b \neq a) \wedge (a \cap b \neq b)$. Os interiores de 'a' e 'b' estão contidos e na mesma dimensão e a intersecção entre 'a' e 'b' é diferente de 'a' e de 'b'.

c) *Cross* (Cruza): $a.Cross(b) \Leftrightarrow [I(a) \cap I(b) \neq \emptyset \wedge (a \cap b \neq a) \wedge (a \cap b \neq b)]$. A intersecção entre os interiores de 'a' e 'b' é diferente de nula e a intersecção entre 'a' e 'b' é diferente de 'a' e de 'b'.

d) *Disjoint* (Disjunto): $a.Disjoint(b) \Leftrightarrow a \cap b = \emptyset$. Não existe intersecção entre 'a' e 'b'.

e) *Equal* (Coincidente): $a.Equals(b) \Leftrightarrow a \subseteq b \wedge b \subseteq a$. Existe coincidência entre 'a' e 'b'.

f) *Within* (Dentro): $a.Within(b) \Leftrightarrow (a \cap b = a) \wedge (I(a) \cap E(b) = \emptyset)$. A intersecção entre 'a' e 'b' é igual a 'a' e a intersecção entre o interior de 'a' e o exterior de 'b' é diferente de nula.

²¹ *Dim* é abreviação de *dimension* e os valores são: dimensão 0 para pontos; 1 para linhas e 2 para polígonos.

2.4 Sistema Geodésico de Referência

Segundo o IBGE²², sistema geodésico de referência é composto por uma figura geométrica representativa da superfície terrestre, posicionada no espaço, permitindo a localização única de cada ponto da superfície em função de suas coordenadas tridimensionais, e materializado por uma rede de estações geodésicas. Em suma, coordenadas como latitude, longitude e altitude, necessitam de um sistema geodésico de referência para sua determinação.

Coordenadas referenciadas a diferentes sistemas geodésicos podem ocasionar problemas topológicos entre os objetos e fenômenos mapeados, como mostra a figura 24:

Figura 24. Exemplo de disjunção entre feições geográficas, referenciadas a diferentes sistemas geodésicos. Fonte: Universidade Federal Fluminense, Sistema Geodésico Brasileiro, Cadastro territorial multifinalitário, 06/11/2012.

O próprio MDC da Prefeitura de São Paulo, por exemplo, identifica as coordenadas UTM (Universal Transversa de Mercator), referenciadas aos DATUM SAD69 e Córrego Alegre, na mesma base em formato CAD, evidenciando o possível deslocamento das feições quando da adoção de diferentes referências geodésicas.

²² Disponível em:

<<http://www.ibge.gov.br/home/geociencias/geodesia/pmrg/faq.shtm#1>>. Acesso em 12 mar. 2015.

Na carta 3313-164, objeto de nossos testes, o deslocamento entre os DATUM é de aproximadamente 38 metros em Y e 18 metros em X. A figura 25 indica o exposto:

Figura 25. Exemplo de coordenadas UTM no MDC da Prefeitura de São Paulo. Notar que as mesmas coordenadas estão deslocadas, devido ao uso de diferentes DATUM.

A biblioteca PROJ.4, como mencionado anteriormente, é a responsável pelos parâmetros dos DATUM e projeções utilizadas pela GDAL/OGR. Estes parâmetros são gerenciados pela biblioteca por meio da classe `OGRSpatialReference`, que possui comunicação com a base de dados EPSG.

Dados essenciais como o elipsóide utilizado, unidades de medida (se graus, metros ou pés), projeção e fuso constam da descrição da referência, como indica, por exemplo, o conteúdo do EPSG 29193 (SAD69, UTM, fuso 23, sul), referente a área da Carta 3313-164, definido pelo OGC:

```
PROJCS["SAD69 / UTM zone
23S",GEOGCS["SAD69",DATUM["South_American_Datum_1969",SPHEROID["GRS 1967
(SAD69)",6378160,298.25,AUTHORITY["EPSG","7050"]],AUTHORITY["EPSG","6618"],PRIMEM["
Greenwich",0,AUTHORITY["EPSG","8901"],UNIT["degree",0.01745329251994328,AUTHORITY["E
PSG","9122"],AUTHORITY["EPSG","4618"],UNIT["metre",1,AUTHORITY["EPSG","9001"],PROJE
CTION["Transverse_Mercator"],PARAMETER["latitude_of_origin",0],PARAMETER["central_meridia
n",-
45],PARAMETER["scale_factor",0.9996],PARAMETER["false_easting",500000],PARAMETER["false
_northing",1000000],AUTHORITY["EPSG","29193"],AXIS["Easting",EAST],AXIS["Northing",NORT
H]]]
```

Para o contexto deste trabalho, analisaremos se as coordenadas espaciais (expressas em metros ou em graus) dos formatos de entrada, serão repassadas de maneira correta aos de saída após as conversões sintáticas. Na hipótese de erro, utilizaremos recursos da GDAL/OGR para efetuar as correções de coordenadas nos dados geoespaciais.

CAPÍTULO 3- FORMATOS DE INTERCÂMBIO DE DADOS GEOESPACIAIS

No universo das conversões sintáticas, o conhecimento da estrutura do formato de dados é de extrema importância para o resultado final da conversão, uma vez que os parâmetros espaciais do formato original só poderão ser transmitidos ao de saída, se existir suporte a eles. Caso contrário, poderiam haver problemas diversos como perdas no componente geométrico dos mapas; mapas com problemas de coordenadas; eliminação de atributos do banco de dados, entre outros.

Conhecendo-se os parâmetros espaciais dos distintos formatos (A, B, C, D... Z), é possível estabelecer regras de comunicação entre eles. Geralmente, este é o processo de conversão sintática entre os formatos padrão para troca de dados entre SIG (como o ESRI Shapefile, MapInfo MIF/MID, AutoCAD DXF, entre outros).

Os formatos²³ escolhidos para análise são representativos de SIG livres e proprietários utilizados no país: TerraView, Spring, QGIS, gvSIG, ArcGIS, MapInfo, ILWIS, Surfer e Google Earth.

3.1 ESRI Shapefile

Este é o formato binário nativo do (antigo) sistema ArcView e foi tornado de uso e implementação livres pela ESRI, ainda que proprietário. É também o formato vetorial padrão de alguns dos SIG *open source* mais importantes na atualidade, como o QGIS, OpenJUMP e o gvSIG, bem como o formato de exportação do sistema ArcGIS.

23 De acordo com o levantamento de 2013 da Safe Software Inc. (empresa desenvolvedora do FME - sistema líder mundial em interoperabilidade geoespacial) os três formatos mais utilizados por usuários do sistema para a conversão de dados geoespaciais entre tecnologias SIG e CAD são: ESRI Shapefile, MicroStation Design (DGN) e AutoCAD DWG/DXF. Mais detalhes em: <<http://blog.safe.com/2014/01/gis-to-cad-conversion/>>. Acesso em 12 mar. 2015.

O formato consiste em pelo menos 3 arquivos principais, como mostra o quadro 1:

Quadro 1. Estrutura do arquivo ESRI shapefile. Adpatado de *FME Readers and Writers* 2012.

Extensão de arquivo	Conteúdo	Arquivo Principal
.shp	Armazena dados geométricos do tipo: pontos, multipontos, poli-linha, polígono e <i>multipatch</i> ²⁴	Sim
.shx	Índice para os dados geométricos	Sim
.dbf	Atributos para os dados geométricos	Sim
.prj	Arquivo para o geoposicionamento dos dados em SIG (possui os parâmetros da projeção e referência geodésica)	Não

Cada arquivo .shp armazena um único tipo de geometria (isto é conhecido como *restrição geométrica*) e existe uma correspondência de um-para-um dos dados geométricos descritos no arquivo .shp com os atributos no arquivo .dbf. Ou seja, cada campo do arquivo .dbf armazena apenas atributos de uma única geometria do arquivo .shp.

Portanto, a conversão de formatos que suportam multi-geometrias para ESRI Shapefile, vai resultar na criação de diversas camadas .shp para cada geometria.

O arquivo .shx funciona como um índice para a consulta nos dados armazenados no arquivo *shapefile*.

²⁴ *Multipatch* é um tipo geométrico basicamente formado por uma coleção de Polygons que podem ser extrudados e se tornarem 3D. Para maiores informações, consultar: <<http://www.esri.com/library/whitepapers/pdfs/multipatch-geometry-type.pdf>>. Acesso em 12 mar. 2015.

O arquivo .prj é opcional, não binário e possui os parâmetros para o geoposicionamento dos *layers* em SIG, como indica o conteúdo de um arquivo .prj referenciado ao EPSG 4326²⁵:

```
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]]
```

O formato não armazena *explicitamente* topologias entre as geometrias.

Pela quantidade de dados armazenados, sistemas que oferecem suporte e tempo de existência, o formato ESRI Shapefile é um dos mais interoperáveis na área do Geoprocessamento, sendo suportado em praticamente todos os sistemas livres e proprietários.

O quadro 2 sintetiza as geometrias suportadas pelo formato:

Quadro 2. Geometrias suportadas pelo formato ESRI Shapefile. Adaptado de *FME Readers and Writers 2012* e *ESRI Shapefile Technical Description: an ESRI White Paper*, 1998.

Geometria	Suporte	Geometria	Suporte
Point e MultiPoint	Sim	Raster	Não
LineString e MultiLineString	Sim	Valores Z	Sim
Polygon e MultiPolygon	Sim	Sólido	Não
Surface	Sim	Nenhuma geometria	Sim
Circles, Elipsis e Arcs	Não	Texto	Sim

²⁵ O EPSG 4326 diz respeito a coordenadas em graus (latitude e longitude) de todo o Planeta, referenciadas ao DATUM WGS84. É a referência geodésica utilizada pelo sistema GPS (*Global Positioning System*).

A ação da biblioteca OGR na conversão sintática para o formato é sintetizada no quadro 3:

Quadro 3. Ação do driver SHP da GDAL/OGR na conversão para o formato ESRI Shapefile. Obs.: o *driver* não suporta o tipo geométrico *Multipatch*.

Geometria de entrada	Ação da biblioteca OGR	Geometria de saída
Point	Produz	Point
Texto	Produz	Point (com <i>label</i> ²⁶)
LineString e MultiLineString	Produz	LineString e MultilineString
LinearRing	Produz	LinearRing
Circles, Elipsis e Arcs	Produz	LineString
Polygon	Produz	Polygon
Surface	Produz	Polygons (quebrados como triângulos)

3.2 MapInfo MIF/MID

Este é o formato vetorial padrão (baseado em texto ASCII²⁷) de intercâmbio de dados do sistema MapInfo e não implementa as restrições geométricas apresentadas pelo ESRI Shapefile, ou seja, um mesmo arquivo .mif pode armazenar diferentes tipos de geometrias.

O quadro 4 sintetiza as geometrias suportadas pelo formato:

²⁶ *Labels* (rótulos) são caracteres alfanuméricos (textos e números) que podem estar associados a geometrias em SIG e CAD.

²⁷ ASCII (*American Standard Code for Information Interchange*) é uma codificação de caracteres baseada no alfabeto inglês em que cada sequência de códigos numéricos na tabela ASCII corresponde a um caractere entre letras, números e símbolos. Os códigos ASCII representam texto em computadores. Fonte: <<http://www.asciitable.com/>>. Acesso em 12 mar. 2015.

Quadro 4. Geometrias suportadas pelo formato MapInfo Mif/Mid. Adaptado de *FME Readers and Writers 2012* e *The MapInfo Interchange File (MIF) Format Specification*, 2009.

Geometria	Suporte	Geometria	Suporte
Point e MultiPoint	Sim	Raster	Não
LineString e MultiLineString	Sim	Sólido	Não
Polygon	Sim	Surface	Não
Arc	Sim	Valores Z	Não
Text	Sim	Nenhuma geometria	Sim

A estrutura do formato está sumarizada no quadro 5:

Quadro 5. Estrutura do arquivo Mif/Mid. Adpatado de *FME Readers and Writers 2012*.

Extensão de arquivo	Conteúdo
.mif	Armazena dados geométricos e os parâmetros do sistema geodésico de referência
.mid	Atributos dos dados geométricos

Sendo nativamente 2D, as geometrias não possuem a capacidade de armazenar dados tridimensionais, a não ser que sejam criados campos de atributos correspondentes. Isto significa que a conversão de pontos cotados ou isolinhas de formatos que armazenam informações 3D (de sistemas CAD, por exemplo), resultam na perda dessas informações no formato de saída.

3.3 AutoCAD DXF

Drawing Exchange Format (DXF) é um formato de intercâmbio de dados do sistema AutoCAD e busca descrever em formato texto (ASCII), o conteúdo do arquivo binário padrão do sistema, o DWG (*Drawing Format*).

Apesar de não possuir suporte a sistemas geodésicos de referência, o DXF é capaz de transmitir e receber coordenadas espaciais (métricas ou em graus).

É composto por algumas seções lógicas, sendo que a mais importante para os processos de conversão sintática é a “*entities*”, na qual estão as feições e suas informações como cor, nome da camada, estilo de linha, geometria e atributos relacionados.

É o formato mais utilizado para a migração de dados de sistemas CAD para SIG.

O quadro 6, sintetiza as geometrias suportadas pelo formato:

Quadro 6. Geometrias suportadas pelo formato AutoCAD DXF. Adaptado de *FME Readers and Writers 2012*.

Geometria	Suporte	Geometria	Suporte
Point e MultiPoint	Sim	Raster	Não
LineString e MultiLineString	Sim	Sólido	Sim
Polygon e MultiPolygon	Sim	Valores Z	Sim
Surface	Sim	Nenhuma geometria	Não
Arc	Sim	Texto	Sim

A OGR suporta a leitura da maioria das versões do formato, mas a escrita apenas na versão AutoCAD 2000. No processo de conversão de outros formatos para o DXF pela biblioteca, o (eventual) conjunto de camadas do primeiro, resultará em apenas um *layer* no segundo, sob o rótulo “*entities*”, o que significa que todo o

conjunto de dados (geometrias, textos, etc) estarão todos em um único *layer* (sem restrição geométrica).

A biblioteca suporta parcialmente o elemento ‘bloco’²⁸, sendo que seus elementos textuais são interpretados como pontos (e *labels* correspondentes) e LineStrings são extraídas para uma única feição chamada “GeometryCollection” (coleção de geometrias do *Simple Features Specification* do OGC).

A documentação técnica indica ainda que cores e espessuras de linhas, bem como orientações dos elementos textuais, buscam ser preservados nas conversões, enquanto estilos de preenchimento não são mantidos.

O quadro 7, sinaliza para alguns aspectos importantes relacionados à conversão para o formato:

Quadro 7. Ação do *driver* DXF da GDAL/OGR na conversão de feições para o formato AutoCAD DXF.

Geometria de entrada	Ação da biblioteca GDAL/OGR	Geometria de saída
Point (sem <i>label</i>)	Produz	Point
Texto	Produz	Ponto com rótulo (<i>label</i>)
LineString e MultiLineString	Produz	LineString e MultiLineString
LinearRing e Polygon	Produz	Polygon (sem preenchimento)
Circles, Elipsis e Arcs	Produz	LineString
Hatch (rachura)	Produz	Polygon (sem preenchimento)

3.4 Google KML

Baseada em linguagem de marcação, ou seja, utilizando o emprego de *tags*, o formato KML (*Keyhole Markup Language*) codifica a transmissão e a representação de dados geoespaciais em SIG e em navegadores geográficos, como

²⁸ Blocos são diversos elementos (geometrias, textos, símbolos) unificados em ambiente CAD, capazes de serem tratados pelo sistema como um único objeto.

o Google Earth, o KDE Marble, entre outros. Em suma, o formato descreve *o que* transmitir e *como* representar um fenômeno geográfico.

O quadro 8, sintetiza o modelo geométrico usado pelo formato:

Quadro 8. Geometrias suportadas pelo formato KML. Adaptado de FME Readers and Writers 2012 e GDAL/OGR Format Specification.

Geometria	Suporte	Geometria	Suporte
Point	Sim	Surface	Sim
LineString	Sim	Raster	Sim
Polygon	Sim	Arc	Não
LinearRing	Sim	Valores Z	Sim
Text	Sim	Nenhuma geometria	Não

A popularização do Geoprocessamento foi facilitada pelo advento de aplicativos como o Google Earth, que permitiram ao público não especializado acesso a imagens de satélites e dados espaciais integrados, além da capacidade de realizar mapeamentos básicos sobre as bases disponíveis. Boa parte dos exploradores 3D, utilizam o formato KML como padrão.

Por isso, uma parte considerável dos dados geoespaciais atualmente encontra-se neste formato - o que explica também a existência dos recursos de importação / exportação em diversos SIG, além da aceitação por parte do OGC do KML (a partir da versão 2.2) como formato padrão de implementação e interoperabilidade em mapas provenientes de dispositivos móveis (2D), exploradores 3D (tipo Google Earth) e servidores de mapas.

3.5 GML

Geography Markup Language (GML) é, assim como o KML, um formato baseado em linguagem de marcação XML para o transporte²⁹ e armazenamento de dados espaciais vetoriais.

Diferentemente de um formato binário, o GML descreve o conteúdo (espacial e não espacial) dos dados de forma *humanamente* inteligível, como mostra o quadro 9:

Quadro 9. Conteúdo de um arquivo vetorial no formato GML.

```
<?xml version="1.0" encoding="utf-8" ?>
<ogr:FeatureCollection
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ogr.maptools.org/ recorte_gml_teste.xsd"
  xmlns:ogr="http://ogr.maptools.org/"
  xmlns:gml="http://www.opengis.net/gml">
  <gml:boundedBy>
    <gml:Box>
      <gml:coord><gml:X>-46,764925</gml:X><gml:Y>-23,812089</gml:Y></gml:coord>
      <gml:coord><gml:X>-46,627381</gml:X><gml:Y>-23,657484</gml:Y></gml:coord>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <ogr:recorte_gml_teste fid="recorte_gml_teste.0">

    <ogr:geometryProperty><gml:Polygon><gml:outerBoundaryIs><gml:LinearRing><gml:coordinates>
-46.667198,-23.684186 -46.666869,-23.684214 -46.666962,-23.684815 -46.667272,-23.68457
-46.667198,-
23.684186</gml:coordinates></gml:LinearRing></gml:outerBoundaryIs></gml:Polygon></ogr:geom
etryProperty>
    <ogr:QUADRAS_ID>8225</ogr:QUADRAS_ID>
    <ogr:ID>8225</ogr:ID>
    <ogr:ID2>795787</ogr:ID2>
    <ogr:QF>17332302</ogr:QF>
    <ogr:Uso_do_Sol></ogr:Uso_do_Sol>
    <ogr:xx>8225</ogr:xx>
    <ogr:cc>8225</ogr:cc>
    </ogr:recorte_gml_teste>
  </gml:featureMember>
```

²⁹ Na área de *Web Mapping*, particularmente, a transmissão de dados vetoriais entre servidores de mapas e aplicativos SIG clientes, é feita por meio do formato GML (geoserviço WFS, *Web Feature Service*).

Notar que são descritas as geometrias, coordenadas geográficas e o conteúdo do banco de dados das feições mapeadas. No caso, estão descritas as feições de polígonos (Polygon), seus limites exteriores (LinearRing) e os atributos correspondentes.

A estrutura do formato GML é sintetizada no quadro 10:

Quadro 10. Modelo geométrico do formato de marcação GML.

Geometria	Suporte	Geometria	Suporte
Point	Sim	Valores Z	Não
Line e LineString	Sim	Raster	Não
Polygon	Sim	Sólido	Não
Arc	Não	Surface	Não
Text	Sim	Nenhuma geometria	Sim

3.6 Microstation DGN

O *Intergraph Standard File Format* (ISFF) ou DGN é um formato vetorial utilizado pelos produtos Intergraph e Bentley e muito disseminado no universo CAD, a ponto de diversos mapas disponíveis para *download* no IBGE, por exemplo, estarem neste formato.

A versão 7, também conhecida como DGN v7, é suportada pela OGR em leitura; o suporte a escrita é limitado.

O modelo geométrico adotado no formato é sintetizado no quadro 11:

Quadro 11. Geometrias suportadas pelo formato DGN.

Geometria	Código	Suporte
Line	3	Sim
LineString	4	Sim

Polygon	6	Sim
Curve	11	Parcial (tratado como linha)
B-Spline	21	Parcial (tratado como linha)
Arc	16	Parcial (tratado como linha)
Elipsis	15	Parcial (tratado como linha)
Text	17	Parcial (tratado como ponto)

Na OGR, o conteúdo do arquivo inteiro é representado por apenas uma camada chamada “*elements*” e o *driver* não reconhece informações sobre os parâmetros do georreferenciamento. Consegue, assim como o DXF, apenas transmitir as coordenadas espaciais nos mapas.

3.7 GeoTIFF

O formato GeoTIFF representa o esforço de mais de 160 organizações ligadas aos campos do Geoprocessamento e Sensoriamento Remoto, para estabelecer um formato padrão para dados *raster* georreferenciados³⁰.

Uma das vantagens da adoção do TIFF³¹, para o contexto do Geoprocessamento e do Sensoriamento Remoto, é sua flexibilidade em permitir a *alteração do cabeçalho* do arquivo, indicando informações além daquelas usuais (dimensão, cores, etc) em uma imagem digital. Assim, informações descrevendo os parâmetros do sistema de referência espacial, por exemplo, podem ser facilmente adicionadas ao arquivo TIFF, transformando-no em um **GeoTIFF**.

O recurso de reprojeção da GDAL (**gdalwarp**) atua no cabeçalho do arquivo GeoTIFF.

O formato possui ainda outras características relevantes como a aplicação de diversos tipos de compressão à imagem e o gerenciamento de arquivos com até

30 Mais detalhes em: <<http://trac.osgeo.org/geotiff/>>. Acesso em 12 mar. 2015.

31 Mais detalhes em: <<http://www.awaresystems.be/imaging/tiff/faq.html>>. Acesso em 12 mar. 2015.

4GB de tamanho (requisito importante para lidar com imagens de satélites e fotografias aéreas).

O GeoTIFF está consolidado como um padrão *de fato*³² na troca de dados entre distintos SIG, sendo o formato matricial padrão no QGIS.

3.8 ArcInfo Grid

A ESRI possui dois formatos *raster*: o binário e proprietário ArcInfo Grid, utilizado dentro do sistema ArcGIS, por exemplo, e; o não proprietário ArcInfo ASCII Grid, usado para exportação de dados matriciais. Como o próprio nome indica, são formatos que se consolidaram com o sistema ArcInfo.

A GDAL possui suporte parcial (somente leitura) para a versão binária e suporte completo (em leitura e escrita) para a versão ASCII.

Como exemplo, o MNT utilizado para ortorretificar as imagens Ikonos de 2002/2003, cobrindo toda a RMSP e disponibilizadas pela Secretaria do Meio Ambiente a várias instituições (entre elas, a USP), está no formato binário ArcInfo Grid. A extensão mais representativa do formato é a “.adf”. As informações do sistema de referência espacial são armazenadas no arquivo “proj.adf”, baseado em ASCII.

Em seguida, empregaremos a GDAL/OGR nos processos de conversão sintática para os dados controlados, selecionados para esta pesquisa.

32 O OGC possui um grupo de trabalho para incorporar o GeoTIFF como padrão de interoperabilidade. Mais detalhes em:
<<http://www.opengeospatial.org/projects/groups/geotiffswg>>. Acesso em 12 mar. 2015.

CAPÍTULO 4 - PROCESSOS DE CONVERSÃO SINTÁTICA ENTRE FORMATOS DE DADOS GEOESPACIAIS

Neste capítulo, serão realizadas as conversões sintáticas nos dados selecionados por meio da biblioteca GDAL/OGR, em sua implementação gráfica e na forma compilada para o sistema operacional. Serão igualmente descritos os procedimentos metodológicos para a solução dos problemas apresentados nos processos de conversão.

A performance da biblioteca será avaliada por meio da conversão de dados geoespaciais:

- 1) da estrutura CAD (formato AutoCAD DXF) para SIG (formatos ESRI Shapefile, MapInfo MIF/MID, Google KML, GML e Microstation DGN);
- 2) da estrutura SIG (formato ESRI Shapefile) para a CAD (formato AutoCAD DXF), e;
- 3) no formato ArcInfo Grid para o GeoTIFF.

Inicialmente, faremos o carregamento dos dados em SIG e CAD, a fim de compararmos as informações fornecidas por estes sistemas, com aquelas provenientes da biblioteca. Se existirem problemas no reconhecimento dos dados por parte dos sistemas, recursos da GDAL/OGR serão empregados na tentativa de eliminá-los.

Em seguida, a biblioteca será empregada na conversão de dados para os formatos escolhidos e os resultados comparados ao estado dos arquivos originais, notadamente em seus aspectos geométricos, topológicos, de atributos e de referência espacial.

Os problemas e as falhas nas conversões serão analisados de forma detalhada.

Finalmente, distintos recursos da biblioteca serão empregados na tentativa de corrigir ou minimizar as falhas nas conversões.

4.1 Leitura e Escrita: AutoCAD DXF para ESRI Shape, MapInfo MIF/MID, Google KML, GML e Microstation DGN.

O DXF e seu correspondente binário (DWG), são formatos estruturados para *desenhos vetoriais*. Entretanto, dada a imensa quantidade de dados geoespaciais nesses formatos, testaremos a performance da OGR na conversão de dados DXF para o universo SIG.

A utilização do OpenJUMP³³ no carregamento dos dados geoespaciais, serve como parâmetro de comparação à biblioteca OGR.

4.1.1) Informações sobre o arquivo 3313-164.DXF:

As informações sobre o arquivo de testes e seu carregamento nos distintos sistemas SIG e CAD objetos da pesquisa, seguem abaixo:

a) Pelo CAD DraftSight:

Este sistema é desenvolvido pela Dassault Systemes, possui uso livre mas código fonte proprietário (*shareware*). Tem a capacidade de ler e escrever em várias versões do formato DXF e DWG, possuindo portanto, uma ótima compatibilidade com o AutoCAD desenvolvido pela Autodesk.

Metadados:

Todos: 694
Simplenote: 316
Bloco: 153
PolyLine: 224
Sombreado: 1

³³ Este SIG implementa recursos próprios e de terceiros (*plug-ins*), para a leitura de dados nos formatos ESRI Shapefile e AutoCAD DXF, respectivamente. Não faz uso da GDAL/OGR.

Figura 26. Folha 3313-164 do MDC 1:1.000, aberta no CAD DraftSight.

A análise dos metadados anteriores, indica que existem 316 elementos textuais, 224 LineStrings, 153 blocos e 1 rachura.

b) Pelo SIG OpenJUMP:

Este sistema suporta a importação do formato DXF versões 8, 14 e 2002, com os seguintes parâmetros e restrições:

- Os elementos geométricos suportados são: Line, Point, Polyline, Polygon, Text e Vertex;
- Elementos que são importados como atributos: Layer, Layer Type, Text, Text_Height, Text_Style, Color (importada apenas como um atributo e, portanto, não pode ser usada para estilização do *layer*);
- Text (elementos textuais) são importados como geometrias de pontos (com os atributos Text e Text_Height);

- Preenchimentos não são suportados, e;
- Rachuras são importadas como linhas contínuas.

É importante enfatizar, novamente, que o OpenJUMP *não* utiliza a GDAL/OGR.

Figura 27. Folha do MDC 3313-164 em CAD, sendo visualizada no SIG OpenJUMP.

As geometrias foram condensadas em um único layer (3313-164). Os elementos da legenda, por sua vez, não são geoposicionados com as demais geometrias do arquivo.

Metadados:

Nome do layer: 3313-164

Número de feições: 5256

Número de pontos: 32622

Tipo de Geometria: Múltiplos tipos de geometria: LineString: 875, Polygon: 3942, Point: 439

Número de atributos: 9

Classe DataSource: DXFFILEReaderWriter

Caminho da fonte: /home/nader/Documentos/Dados/3313-164.DXF

```
T:Extent: T:xMin:-58.77000000001864  
          T:yMin:-44.99000000002235  
          T:xMax:324719.5703000001  
          T:yMax:7393727.790721915
```

Os metadados acima informam a existência de mais feições daquelas informadas no DraftSight.

Como este SIG reconhece o elemento Vertex, é possível que isto explique o excessivo número de pontos do arquivo; ou seja, estes seriam vértices dos vários segmentos de retas componentes das LineStrings.

c) Pelo QGIS 1.8:

A capacidade de abrir e visualizar arquivos no QGIS é dada pela GDAL/OGR compilada para o SIG. Esta compilação utiliza a mesma versão da biblioteca em que estamos fazendo nossos testes na linha de comando, porém de uma forma gráfica.

Dado que a expressão mais fundamental do mapa é a visual, o QGIS se torna um recurso interessante para avaliarmos o estado dos arquivos antes e pós-conversão pela biblioteca.

Segundo a documentação técnica, a GDAL/OGR suporta a leitura da maioria das versões do formato DXF, e a escrita na versão AutoCAD 2000.

Figura 28. Folha do MDC 3313-164 em CAD, sendo visualizada no QGIS 1.8.

Notar que as geometrias de LineString e Polygon não foram apresentadas, somente as de pontos (que contêm *labels*). O sistema apresenta suporte parcial aos atributos relacionados às feições CAD. É importante dizer que as bibliotecas GDAL/OGR são as responsáveis pela abertura de arquivos vetoriais e matriciais no QGIS.

Geral:

Tipo de armazenamento dessa camada: DXF

Fonte para esta camada: /home/nader/Documents/Dados/3313-164.DXF

Tipo de geometria das feições nesta camada: Point

Número de feições nesta camada: 844

Editar capacidades para esta camada:

Extensão:

Sistema de unidades de referência espacial na camada : xMin,yMin 323989.07, 7393097.04 :xMax, yMax 324775.57, 7393761.61

Sistema de referência espacial da camada:

+proj=longlat +ellps=aust_SA +towgs84=-57,1,-41,0,0,0,0 +no_defs

Os pontos do mapa são o resultado da interpretação pela GDAL/OGR dos elementos textos do CAD. As demais geometrias não foram reconhecidas.

d) Pela OGR (linha de comando):

Como mencionado, esta biblioteca suporta a leitura do formato DXF na maioria de suas versões e a escrita na versão AutoCAD 2000.

Metadados:

ogrinfo -al 3313-164.DXF

```
INFO: Open of `3313-164.DXF'
      using driver `DXF' successful.
```

```
Layer name: entities
```

```
Geometry: Unknown (any)
```

```
Feature Count: 844
```

```
Extent: (323989.069200, 7393097.039300) - (324775.569200, 7393761.609300)
```

```
Layer SRS WKT:
```

```
(unknown)
```

```
Layer: String (0.0)
```

```
SubClasses: String (0.0)
```

```
ExtendedEntity: String (0.0)
```

```
Linetype: String (0.0)
```

```
EntityHandle: String (0.0)
```

```
Text: String (0.0)
```

Como a documentação técnica da biblioteca indica, todos os *layers* do arquivo são condensados numa única camada chamada 'entities', que foi reconhecida.

A biblioteca reconhece a extensão espacial (em coordenadas UTM) e o número de feições mas não as distingue geometricamente no conjunto do *layer*.

e) Análise do carregamento do arquivo 3313-164.DXF em SIG e CAD:

Cada SIG apresentou informações diferentes sobre as estruturas geométricas, de atributos e referência espacial existentes no arquivo DXF.

O formato DXF não armazena topologia, apesar do arquivo analisado ter sido gerado por meio de *planar enforcement*.

Os metadados mais detalhados foram apresentados pelo sistema OpenJUMP. Enquanto este informa a existência de múltiplas geometrias, o QGIS lista apenas a existência de pontos na camada. O primeiro, contudo, não consegue gerenciar a informação do referencial geodésico e projeção de maneira satisfatória. O QGIS, por sua vez, permite a escolha da projeção e referencial geodésico para o mapa por meio do código EPSG e apresenta a extensão das coordenadas UTM. O OpenJUMP lista parcialmente as coordenadas UTM, mas em conjunto com outras em graus de latitude e longitude diferentes da área a que se refere o mapa. Isto pode indicar erro do sistema na atribuição automática do fuso da projeção.

A biblioteca OGR, por sua vez, não consegue reconhecer as geometrias condensadas sob o *layer* “entities”, somente sua extensão espacial em coordenadas UTM.

4.1.1.1) Análise de conteúdo parcial do arquivo 3313-164.DXF:

Uma vez que o CAD e os SIG apresentaram informações distintas sobre o mesmo arquivo, isolamos uma feição para analisar como os sistemas gerenciavam o dado parcial, ou seja, apenas parte do conteúdo do arquivo. A representação escolhida foi a de curvas de nível (isolinhas), como indica a figura 29:

a) Pelo DraftSight:

Figura 29. Foram isolados apenas os *layers* de Curvas mestras e intermediárias.

Notar a quebra entre as linhas (para a inserção de labels com a indicação de altitude no CAD) e como alguns elementos que não são representações de isolinhas (especialmente o círculo no canto superior esquerdo do mapa, que indica uma praça dentro da praça do relógio na USP), fazem parte da camada. Isso indica a mistura de objetos semanticamente distintos no mesmo layer, comprometendo a qualidade da base cartográfica. Este é um problema que acompanha o arquivo original.

Das feições apresentadas, selecionamos a *LineString* “600_CURVA_MESTRA”, que possui os seguintes metadados:

```
LWPOLYLINE
Layer
600_CURVA_MESTRA
Área
0
Perímetro
624.968
Local 0
X=324537.174 Y=7393189.237 Z=725
Local 1
X=324536.758 Y=7393189.528 Z=725
```

Local 2
X=324535.366 Y=7393190.077 Z=725
.....
Local 15
X=324528.592 Y=7393204.439 Z=725
Local 16
X=324528.435 Y=7393205.658 Z=725
Local 17
X=324527.873 Y=7393206.946 Z=725
Local 18
X=324527.298 Y=7393208.345 Z=725
Local 19
X=324526.676 Y=7393209.54 Z=725

Foram apresentados no total 20 valores (incluindo o valor 'Local 0'), identificados pelas coordenadas UTM e o identificador de altitude 'Z', (sumarizados nos parâmetros acima). Em ambiente CAD, a LineString (chamado pelo CAD de LWPOLYLINE) pode receber valores 3D.

b) Pelo OpenJUMP:

Figura 30. Representação de Curvas de Nível em formato DXF carregadas no OpenJUMP.

As geometrias de linhas foram preservadas mas a geometria de LinearRing (no canto superior esquerdo do mapa, que representa a praça), foi transformada em Polygon.

A mesma feição no OpenJUMP apresenta os seguintes metadados:

3313-164_Isolinhas Isoladas

```
FID 55 LINESTRING (324537.1737827495 7393189.236701082 725, 324536.758356899
7393189.527659858 725, 324535.3658588119 7393190.077083993 725, 324534.7936152665
7393190.341199137 725, 324534.4118155613 7393190.517416209 725, 324533.5634015538
7393191.191516159 725, 324531.8100867506 7393193.584346342 725, 324529.8844558329
7393196.02568936 725, 324528.9823690952 7393197.421191247 725, 324528.3643058264
.....
7393270.799999998 725, 324217.3199999966 7393271.669999998 725, 324214.3799999989
7393273.199999997 725, 324213.0199999996 7393274.049999998 725, 324212.5399999991
7393274.389999999 725, 324211.4899999983 7393274.959999999 725, 324210.2731622619
7393275.799464008 725) LAYER:

600_CURVA_MESTRA LTYPE: BYLAYER ELEVATION: 725.0 THICKNESS: 0.0 COLOR: 256
TEXT: TEXT_HEIGHT: TEXT_ROTATION: TEXT_STYLE:
```

Notar que o OpenJUMP consegue reconhecer a geometria (indicada por LINESTRING), as coordenadas UTM e o valor da altitude 'Z' que, nos metadados, aparece como 'ELEVATION' (sumarizados nos parâmetros acima).

c) Pelo QGIS 1.8:

Figura 31. Camadas de isolinhas carregadas no QGIS.

Notar que as geometrias (LineString) são preservadas e a praça não foi interpretada como Polygon (mas como LinearRing). Os atributos vinculados à feição selecionada (notadamente o valor de altitude 'Z') foram perdidos.

A feição, interpretada como ponto pelo sistema, possui os seguintes atributos:

```
Layer: 600_CURVA_MESTRA
SubClasses: AcDbEntity:AcDbPolyline
ExtendedEntity:
Linetype:
EntityHandle: 152
Text:
```

d) Análise da leitura parcial do arquivo pela biblioteca e sistemas SIG e CAD:

O QGIS identifica a feição como linha, tendo por subclasse a *polyline*. Entretanto, a LineString que era originalmente 3D, perdeu seu atributo de altitude no SIG.

No OpenJUMP, houve um reconhecimento similar da feição em comparação ao DraftSight (apenas 1 vértice a mais gerado na leitura do arquivo). Este sistema reconhece os valores de elevação das isolinhas.

Com relação ao georreferenciamento, ambos os SIG apresentaram corretamente as coordenadas UTM da feição.

4.1.1.2) Reescrita do arquivo MDC 3313-164 para a versão AutoCAD 2000

A análise anterior levanta a seguinte questão: será que a OGR no QGIS, ao contrário do OpenJUMP, não conseguiu interpretar o formato CAD por falta de suporte à versão? Uma vez que desconhecemos qual a versão do DXF e em que sistema CAD o arquivo foi criado, esta é uma possibilidade real.

Para respondermos a isso, reescrevemos o arquivo 3313-164 no DraftSight para o formato DXF versão R2000-2002 que é suportada pela biblioteca.

Em seguida, abrimos o arquivo convertido no QGIS, como ilustra a figura 32:

Figura 32. Arquivo do MDC convertido pelo DraftSight para formato DXF versão 2000-2002.

Notar que foram obtidas as mesmas geometrias (Points) quando do carregamento do arquivo CAD original.

Metadados:

Geral:

Tipo de armazenamento dessa camada: DXF

Fonte para esta camada: /home/nader/Documents/Dados/3313-164_ver2000.dxf

Tipo de geometria das feições nesta camada: Point

Número de feições nesta camada: 844

Editar capacidades para esta camada:

Extensão:

Sistema de unidades de referência espacial na camada : xMin,yMin 323989.07, 7393097.04 :xMax, yMax 324775.57, 7393761.61

Sistema de referência espacial da camada:

+proj=utm +zone=23 +south +ellps=aust_SA +towgs84=-57,1,-41,0,0,0,0 +units=m

+no_defs

Como demonstrado, a OGR implementada no QGIS não conseguiu ler completamente o arquivo 3313-164 do MDC, mesmo após sua conversão para a versão R2000-2002 do formato DXF.

Por nossa experiência com o MDC da Prefeitura de São Paulo, sabemos que o QGIS consegue abrir muitas folhas, enquanto não reconhece outras. Este é um problema desconhecido e atinge cartas aleatórias.

Para compreendemos a situação pós-conversão, utilizaremos dois recursos no Linux, capazes de comparar códigos baseados em texto³⁴. O comando é:

```
diff -U 3 -dHrN -- 3313-164.DXF 3313-164_ver2000.dxf
```

e sua resposta gráfica (aplicativo Kompare) é mostrada na figura 33:

Figura 33. Aplicativo Kompare, que implementa o recurso Diff graficamente, mostrando a comparação entre os arquivos 3313-164 original e o convertido para a versão 2000-2002 do DXF.

Com este aplicativo, é possível comparar as mesmas linhas de código nos dois arquivos distintos.

³⁴ Esta comparação é possível porque ambas as versões estão no formato DXF, baseado em texto ASCII.

Os metadados gerados para os dois arquivos, estão a seguir:

Arquivo antigo: 3313-164.DXF
Arquivo novo: 3313-164_ver2000.dxf
Formato: Unificado
Número de blocos: 6.991
Número de diferenças: 4.676

O indicador acima mostra que o DraftSight alterou significativamente o arquivo após a conversão para a versão 2000-2002 do formato DXF, mas isto não foi suficiente para que a OGR implementada no QGIS interpretasse corretamente o novo arquivo gerado.

A comparação dos *cabeçalhos* e *códigos finais* dos arquivos mostra que ambos estão iguais, o que indica que somente a análise minuciosa do código poderia, eventualmente, demonstrar o motivo do problema - o que fugiria do escopo deste trabalho.

4.1.1.3) Carregamento dos dados na nova versão do QGIS (2.0.1)

Durante a escrita desta pesquisa, uma nova versão do QGIS (2.0.1, Dufour) foi lançada. As duas versões do QGIS (1.8 e 2.0.1) utilizadas, estão rodando sobre a mesma versão 1.10.0 da GDAL/OGR.

Para identificarmos se o problema está mesmo na biblioteca ou se é um problema do QGIS para carregamento de arquivos DXF, utilizamos a versão 2.0.1 do SIG, para efeitos de comparação com a versão anterior, como mostram a figura 34 e os metadados a seguir:

Figura 34. Arquivo CAD 3313-164 carregado no QGIS versão 2.0.1.

Notar que, mesmo rodando sobre a mesma versão da GDAL/OGR da versão anterior do SIG, desta vez o sistema consegue interpretar de maneira satisfatória a maior parte do conteúdo dos dados, com exceção de algumas geometrias e textos da legenda.

Metadados:

```
entities Polygon:1
entities GeometryCollection:0
entities LineString: 375
entities Point: 467
```

Nesta versão, diferentemente da 1.8, o sistema apresenta os seguintes recursos para a importação do arquivo DXF, como indica a figura 35:

Figura 35. Módulo de importação do formato DXF no QGIS.

Notar que o QGIS 2 apresenta o recurso (caixa de diálogo) de separação de geometrias contidas no layer 'entities' (que condensa todas as geometrias e atributos no formato DXF), o que não acontecia na versão 1.8.

O *layer* de GeometryCollection quando importado no QGIS, não apresentou feição.

O Polygon foi importado como uma geometria pequena, um elemento componente da legenda do mapa (esta que foi quase toda perdida na importação, com exceção de parte dos elementos textuais).

O QGIS indicou corretamente as coordenadas da projeção UTM.

As isolinhas perderam seus valores de altitude.

4.1.1.4) Interpretação do arquivo 3313-164.DXF pelo *OGR Virtual Format (VRT)*

É possível afirmar que a OGR implementada no QGIS 1.8 não tenha reconhecido o conteúdo do arquivo CAD porque não conseguiu interpretar e separar as diferentes geometrias condensadas no *layer* 'entities'. O que reforça esta ideia é a capacidade do QGIS em reconhecer parte do conteúdo (o layer de Curvas de Nível isolado, por exemplo), mas não o arquivo todo.

Uma alternativa a essa limitação, é usar a classe de interpretação de geometrias (OGRGeometry) presente no formato virtual OGR VRT. Os códigos abaixo irão extrair informações geométricas de Points, LineString, Polygon e

GeometryCollection³⁵ do arquivo DXF, e transferí-los para o formato virtual OGR VRT, com a mesma referência espacial do arquivo CAD original (SAD-69, fuso 23 sul, da projeção UTM, identificados no código como '*<LayerSRS>EPSG:29193</LayerSRS>*'). Uma vez que os elementos textuais do CAD são interpretados como Points, esta é uma forma interessante de recuperar os atributos do formato DXF para dentro do SIG.

Se bem sucedido este processo, será possível recuperar graficamente no QGIS as geometrias e textos do arquivo CAD original.

Seguem os códigos do formato virtual OGR VRT:

Layer de Point:

```
<OGRVRTDataSource>
<OGRVRTLayer name="3313-164_pontos">
<SrcDataSource relativeToVRT="1">3313-164.DXF</SrcDataSource>
<GeometryType>wkbPoint</GeometryType>
<LayerSRS>EPSG:29193</LayerSRS>
<SrcSQL>SELECT * FROM entities WHERE OGR_GEOMETRY='POINT'</SrcSQL>
</OGRVRTLayer>
</OGRVRTDataSource>
```

Layer de LineString:

```
<OGRVRTDataSource>
<OGRVRTLayer name="3313-164_linhas">
<SrcDataSource relativeToVRT="1">3313-164.DXF</SrcDataSource>
<GeometryType>wkbLineString</GeometryType>
<LayerSRS>EPSG:29193</LayerSRS>
<SrcSQL>SELECT * FROM entities WHERE
OGR_GEOMETRY='LINESTRING'</SrcSQL>
</OGRVRTLayer>
</OGRVRTDataSource>
```

35 Deste layer, as geometrias extraídas serão transformadas em LineStrings e os textos em Points com *labels*, por implementação da biblioteca.

Layer de Polygon:

```
<OGRVRTDataSource>
<OGRVRTLayer name="3313-164_poligonos">
<SrcDataSource relativeToVRT="1">3313-164.DXF</SrcDataSource>
<GeometryType>wkbPolygon</GeometryType>
<LayerSRS>EPSG:29193</LayerSRS>
<SrcSQL>SELECT * FROM entities WHERE OGR_GEOMETRY='POLYGON'</SrcSQL>
</OGRVRTLayer>
</OGRVRTDataSource>
```

Layer de GeometryCollection

```
<OGRVRTDataSource>
<OGRVRTLayer name="3313-164_geometry_collection">
<SrcDataSource relativeToVRT="1">3313-164.DXF</SrcDataSource>
<GeometryType>wkbGeometryCollection</GeometryType>
<LayerSRS>EPSG:29193</LayerSRS>
<SrcSQL>SELECT * FROM entities WHERE
OGR_GEOMETRY='LINESTRING'</SrcSQL>
</OGRVRTLayer>
</OGRVRTDataSource>
```

Para que a extração funcione, é preciso que os arquivos .VRT estejam na mesma pasta em disco do arquivo CAD a ser analisado pela biblioteca.

Para cada arquivo OGR VRT criado, acessamos suas informações pela OGR na linha de comando:

Layer de Point:

```
ogrinfo -so 3313-164_pontos.vrt
INFO: Open of `3313-164_pontos.vrt'
      using driver `VRT' successful.
1: 3313-164_pontos (Point)
```

Layer de LineString:

```
ogrinfo -so 3313-164_linhas.vrt  
INFO: Open of `3313-164_linhas.vrt'  
      using driver `VRT' successful.  
1: 3313-164_linhas (Line String)
```

Layer de Polygon:

```
ogrinfo -so 3313-164_poligonos.vrt  
INFO: Open of `3313-164_poligonos.vrt'  
      using driver `VRT' successful.  
1: 3313-164_poligonos (Polygon)
```

Layer de GeometryCollection

```
ogrinfo -so 3313-164_geometry_collection.vrt  
INFO: Open of `3313-164_geometry_collection.vrt'  
      using driver `VRT' successful.  
1: 3313-164_geometry_collection (Geometry Collection)
```

As informações acima mostram que a leitura dos arquivos por parte da biblioteca no formato virtual OGR VRT, foi bem sucedida.

Feito isto, basta carregar os arquivos OGR VRT no QGIS. A figura 36 ilustra o carregamento dos dados:

Figura 36. Carregamentos do arquivos gerados pelos códigos no formato OGR VRT no QGIS.

Notar que as informações geométricas da legenda foram perdidas. Só é possível carregar dados neste formato no QGIS, porque este SIG é desenvolvido sobre a própria GDAL/OGR.

Metadados:

```
Points: 467
LineString: 375
Polygon: 1
GeometryCollection: 0
```

Os metadados são coincidentes com aqueles gerados no carregamento do arquivo 3313-164.DXF no QGIS 2.0.1.

O *layer* de GeometryCollection, apesar de referenciado no SIG, está vazio. Ou seja, o OGR VRT com a separação por geometrias não conseguiu interpretar elementos do bloco CAD.

O sistema apresentou corretamente as coordenadas UTM dos mapas.

As isolinhas perderam seus valores de altitude no SIG.

4.1.1.5) Reescrita do arquivo 3313-164 para o formato DXF: recurso ‘explodecollections’

Pelo que investigamos até aqui, é provável que a maior dificuldade de interpretação desta carta pela biblioteca, esteja relacionada aos elementos ‘blocos’ utilizados no arquivo DXF. Uma forma de resolver o problema e, ao mesmo tempo, recuperar os elementos da legenda (condensados no bloco), é gerar um novo arquivo CAD no mesmo formato DXF, ‘explodindo’³⁶ os blocos existentes (unidos no *layer* GeometryCollection) para que a GDAL/OGR produza uma nova feição para cada geometria da coleção. Se bem sucedido este recurso, mais elementos poderão ser recuperados graficamente em SIG. O comando é:

```
ogr2ogr -f "DXF" -explodecollections geo.dxf 3313-164.DXF
```

e o resultado é ilustrado na figura 37:

Figura 37. Resultado da ‘explosão’ do arquivo CAD 3313-164 e a geração do novo arquivo DXF ‘geo.dxf’.

Notar que foram gerados 3 *layers* ‘entities’, um para cada tipo de geometria (LineString, Polygon e Point).

36 ‘Explodir’ elementos no CAD, significar desagrupar textos e geometrias em suas partes simples (individuais).

Metadados do arquivo geo.dxf gerado com o recuso 'explodecollection':

Polygon: 1

LineString: 4754

Point: 467

Tamanho em disco: 2,1MB

Podemos perceber, visualmente, que o resultado da geração de um novo arquivo CAD por meio do recurso '**explodecollection**' é bem superior ao recurso nativo de abertura do arquivo pela OGR no QGIS 2.0.1.

Após o emprego do recurso, as geometrias no *layer* GeometryCollection foram distribuídas para o *layer* de LineString; nas geometrias de Points da legenda, os elementos textuais não foram recuperados.

O QGIS recebeu com êxito as coordenadas UTM.

O número de LineStrings aumentou, devido a recuperação dos diversos elementos da legenda que antes eram perdidos na importação.

As isolinhas perderam seus valor de altitude no QGIS.

O número de Points se manteve o mesmo. Isto pode indicar que: 1) a biblioteca não conseguiu recuperar os demais pontos existentes, ou; 2) não existem mais pontos a serem recuperados.

É interessante notar que o novo arquivo DXF gerado pela OGR, fica muito próximo - visualmente - do original quando aberto no DraftSight. Com exceção de alguns problemas nos elementos da legenda (perda de textos e preenchimento da barra de escala gráfica; alteração na codificação dos textos, notadamente caracteres especiais como 'ç'; substituição do tipo de fonte e cores de alguns símbolos), o mapa conservou seu conteúdo geométrico.

É importante dizer que, aberto no DraftSight, o arquivo reescrito pela biblioteca manteve os valores de altitude das isolinhas.

De um total de 97 *layers* originais, apenas 27 foram conservados. Esta redução se deu porque distintos *layers* de textos foram unificados.

Para compararmos o arquivo 3313-164.dxf (após a explosão dos blocos no CAD), com o geo.dxf (gerado pelo recurso **‘explodcollection’** da GDAL/OGR), indicamos os metadados do primeiro, conforme segue:

Metadados do arquivo 3313-164.dxf com todos os blocos ‘explodidos’ no CAD:

Todos: 12711
 Simplenote: 462
 PolyLine2D: 3
 Arco: 1
 Linha: 8132
 Sólido: 19
 Elipse: 13
 Ranhura: 4
 Nota: 1
 PolyLine: 4074
 Imagem: 2
 Tamanho ocupado em disco: 6,9MB.

A comparação dos metadados de ambos os arquivos, *somente*, indica que a conversão não foi bem sucedida. Contudo, quando analisamos visualmente os mapas, lado a lado, podemos perceber que a interpretação do DXF pela biblioteca foi satisfatória, conforme indica a figura 38:

Figura 38. Comparação do arquivo gerado pela GDAL/OGR após a utilização do recurso **‘explodcollection’** (à esquerda) com o original 3313-164.dxf (à direita), no Draftsight.

Adicionalmente, para sabermos se a perda dos valores de altitude das isolinhas se deve a um problema da biblioteca ou se é uma deficiência do QGIS, utilizamos o seguinte comando da OGR para a leitura detalhada do *layer* '600_CURVA_MESTRA' representando uma isolinha de altitude. O resultado é apresentado em seguida:

```
ogrinfo -where "Layer='600_CURVA_MESTRA'" geo.dxf entities
```

```
600_CURVA_MESTRA
```

```
OGRFeature(entities):215
```

```
Layer (String) = 600_CURVA_MESTRA
```

```
SubClasses (String) = AcDbEntity:AcDbPolyline
```

```
ExtendedEntity (String) = (null)
```

```
Linetype (String) = (null)
```

```
EntityHandle (String) = 200D7
```

```
Text (String) = (null)
```

```
Style = PEN(c:#7f0000)
```

```
LINESTRING (324077.193413726985455 7393377.816042349673808 725,324076.61
7393378.02 725,324074.519999999029096 7393378.63 725,324073.4399999998023268 7393379.0
725,324072.36 7393379.59 725,324071.3199999996980187 7393380.25
725,324070.199999999022111 7393380.89 725,324069.219999999898253 7393381.24
725,324066.8899999996987171 7393381.77 725,324065.6399999996987171 7393381.87
725,324064.079999999797903 7393381.86 725,324063.0399999998989515 7393381.81
725,324061.77 7393381.66 725,324060.6199999997026753 7393381.75
725,324059.5499999997019768 7393382.06 725,324058.34 7393382.76 725,324057.36
7393383.57 725,324056.4399999998023268 7393384.29 725,324055.59 7393384.83
725,324054.5999999997997656 7393385.49 725,324053.3799999999015126 7393387.8
725,324052.1899999998023268 7393389.37 725,324050.1699999998004641 7393391.96
725,324048.4799999997012783 7393394.37 725,324047.86 7393395.48
725,324047.1699999998004641 7393398.11 725,324047.0199999999029096 7393399.13
725,324046.7799999997001141 7393401.2 725,324046.6699999998004641 7393402.85
725,324046.4199999998004641 7393403.99 725,324046.3999999998975545 7393405.39
725,324046.75 7393406.93 725,324047.079999999797903 7393408.49
725,324047.3799999999015126 7393410.07 725,324047.4599999996994156 7393412.68
725,324047.3899999996987171 7393415.15 725,324047.0999999997997656 7393417.79
725,324046.8499999997997656 7393418.88 725,324046.0999999997997656 7393420.59
725,324045.4199999998004641 7393421.86 725,324044.86 7393423.23
725,324044.442870970990043 7393424.349126660265028 725)
```

Os valores acima mostram que a biblioteca, além das coordenadas na projeção UTM, reconhece o valor de altitude da isolinhas (LineStrings), representado pelo número **725** (após as coordenadas).

Point: 467
Polygon: 0

Percebemos uma melhora considerável em relação aos elementos textuais. Apesar dos metadados indicarem que o número de pontos (consequentemente, também o de textos) permaneceu inalterado, desta vez conseguimos visualizar mais itens textuais da legenda. Isto significa que, nos processos anteriores, os textos eram interpretados pela biblioteca, mas o QGIS não conseguia recuperá-los graficamente.

Novamente, as coordenadas UTM foram corretamente indicadas.

A eliminação dos blocos fez com que a biblioteca não conseguisse interpretar o *layer* de Polygon. Esta geometria foi perdida.

O número excessivamente maior de LineStrings indica que estas foram quebradas em pedaços menores (segmentos de retas), fruto da eliminação dos blocos CAD.

As isolinhas perderam seus valores de altitude no QGIS e foram mescladas com outras geometrias LineString no *layer* 'entities LineString'.

4.1.1.7) Conclusão do carregamento dos arquivo DXF pela OGR em CAD e SIG

Comparando as versões do QGIS - que rodam sobre a mesma compilação 1.10.0 da biblioteca - podemos concluir que a implementação desta é consideravelmente melhor na última versão do SIG.

Quando abrimos o arquivo 3313-164.DXF no QGIS 2.0.1, o sistema apresentou um menu para a seleção das geometrias (Point, LineString, Polygon e GeometryCollection) que poderiam ser inseridas como *layers*; o que não aconteceu com a versão 1.8 (que listou apenas as geometrias de Point e seus *labels* respectivos).

Como vimos, o carregamento de dados CAD em SIG pela OGR é, sem dúvida, melhorado se for possível separá-los por tipos de geometrias e explodí-las (*explodecollections*). Este recurso foi superior àqueles implementados graficamente

no QGIS 2.0.1 e à utilização do formato virtual OGR VRT. A separação de geometrias e sua consequente explosão, pode minimizar o problema do não reconhecimento das múltiplas geometrias condensadas no *layer* 'entities', após sua importação em SIG. De igual modo, a eliminação dos blocos do arquivo CAD resulta em uma melhora significativa na qualidade do mapa e sua recuperação gráfica no QGIS. Isto confirma que o suporte da biblioteca ao elemento 'bloco' não é completo e pode prejudicar o carregamento de arquivos CAD (que usam este recurso) no QGIS e sua interpretação pela OGR.

Notamos que as isolinhas no QGIS não tiveram seus valores de altitudes reconhecidos, mas estes se conservaram em ambiente CAD, mesmo durante o processo de reescrita do arquivo CAD pela OGR. Isto indica que a biblioteca conserva os valores associados às geometrias (LineStrings, neste caso) na interpretação do arquivo, mas existe uma deficiência de sua leitura no QGIS. Para avaliarmos se era realmente um problema da biblioteca ou uma deficiência do sistema, abrimos uma feição do arquivo reescrito pela OGR no OpenJUMP, o qual listou corretamente os valores de altitude das isolinhas. A própria biblioteca, pelo recurso **ogrinfo**, apresentou corretamente os valores.

A OGR conseguiu interpretar e transmitir corretamente as coordenadas UTM dos arquivos CAD.

O formato DXF e seu correspondente binário (DWG), trazem uma dificuldade intrínseca pelo fato de serem estruturados para desenhos vetoriais, não dados geoespaciais e suas necessidades específicas. Estes dados, como vimos, exigem estruturas próprias de armazenamento que contemplem aspectos como topologia, atributos e referência espacial - parcialmente suportados no universo CAD.

Apesar disso, em nossos testes, identificamos que a biblioteca - com perdas principalmente nos elementos textuais da legenda que estavam condensados no bloco CAD - conseguiu interpretar a maior parte dos elementos geométricos e textuais da base DXF. A reescrita do arquivo CAD para o formato DXF pela OGR, mostrou uma similaridade muito grande com o arquivo original (como a preservação de cores, orientação dos textos, recuperação de geometrias e coordenadas UTM).

Concluimos, portanto, que a OGR pode ser amplamente usada na leitura de dados CAD, principalmente utilizando os recursos avançados do aplicativo **ogr2ogr**.

4.1.2) Conversão de AutoCAD DXF para ESRI Shapefile

O formato ESRI shapefile é um dos mais interoperáveis em SIG, sendo suportado pela maioria dos sistemas livres e proprietários existentes.

O sucesso no processamento e conversão do CAD DXF para o formato de teste, pode significar um ganho efetivo de interoperabilidade geoespacial, já que um dos grandes desafios de integração de dados no Geoprocessamento, consiste em migrar dados CAD para formatos SIG.

A conversão da carta 3313-164.DXF para o formato ESRI, é feita pelo seguinte comando:

```
ogr2ogr -f "ESRI Shapefile" 3313-164.shp 3313-164.DXF
```

Resultado:

```
Warning 6: Normalized/launched field name: 'ExtendedEntity' to 'ExtendedEn'
Warning 6: Normalized/launched field name: 'EntityHandle' to 'EntityHand'
ERROR 1: Attempt to write non-point (LINESTRING) geometry to point shapefile.
ERROR 1: Terminating translation prematurely after failed
translation of layer entities (use -skipfailures to skip errors)
```

a) Informações sobre o arquivo criado pela GDAL/OGR:

Comando:

```
ogrinfo -al 3313-164.shp
```

```
INFO: Open of `3313-164.shp'
      using driver `ESRI Shapefile' successful.
```

```
Layer name: 3313-164
Geometry: 3D Point
```

```

Feature Count: 0
Extent: (0.000000, 0.000000) - (0.000000, 0.000000)
Layer SRS WKT:
(unknown)
Layer: String (80.0)
SubClasses: String (80.0)
ExtendedEn: String (80.0)
Linetype: String (80.0)
EntityHand: String (80.0)
Text: String (80.0)

```

A conversão apresentou erros, o que demonstra os consoles de informações abaixo:

“ERROR 1: Terminating translation prematurely after failed translation of layer entities (use -skipfailures to skip errors)” e “Feature Count = 0”.

O arquivo ESRI Shapefile foi criado mas está vazio, sem conteúdo e não pode ser visualizado no OpenJUMP e QGIS.

Uma forma de contornar o problema é, teoricamente, separar o arquivo por conjunto de geometrias, fazendo uso da *restrição geométrica* utilizada pelo formato ESRI Shapefile em conjunto com a classe **OGRGeometry**:

a) Extração para camada de Pontos:

```

ogr2ogr 3313-164_points.shp 3313-164.DXF -where "ogr_geometry = 'point'"
Warning 6: Normalized/laundered field name: 'ExtendedEntity' to 'ExtendedEn'
Warning 6: Normalized/laundered field name: 'EntityHandle' to 'EntityHand'

```

b) Extração para camada de Linhas:

```

ogr2ogr 3313-164_lines.shp 3313-164.DXF -where "ogr_geometry = 'linestring'"
Warning 6: Normalized/laundered field name: 'ExtendedEntity' to 'ExtendedEn'
Warning 6: Normalized/laundered field name: 'EntityHandle' to 'EntityHand'

```


c) Extração para camada de Polígonos:

```
ogr2ogr 3313-164_polygon.shp 3313-164.DXF -where "ogr_geometry = 'polygon'"
```

Warning 6: Normalized/laundered field name: 'ExtendedEntity' to 'ExtendedEn'

Warning 6: Normalized/laundered field name: 'EntityHandle' to 'EntityHand'

d) Extração para camada de GeometryCollection (que poderá resultar em Points e LineStrings):

```
ogr2ogr 3313-164_geometry_collection.shp 3313-164.DXF -where "ogr_geometry = 'GeometryCollection'"
```

Warning 6: Normalized/laundered field name: 'ExtendedEntity' to 'ExtendedEn'

Warning 6: Normalized/laundered field name: 'EntityHandle' to 'EntityHand'

A visualização no QGIS³⁷ da conversão de DXF para ESRI Shapefile, pela ação do aplicativo **ogr2ogr**, é mostrado na figura 40 e nos metadados a seguir:

Figura 40. Resultado da conversão do arquivo 3313-164 de DXF para ESRI Shapefile, utilizando a GDAL/OGR.

A ação da biblioteca foi analisar o arquivo CAD por geometrias e criar um *layer* para cada tipo geométrico.

³⁷ Como a versão 1.8 do sistema apresentou muitos problemas na implementação gráfica da GDAL/OGR, a partir daqui utilizaremos a versão 2.0.1 em nossos testes.

Metadados:

Polygon:1

Lines: 375

Points: 467

GeometryCollection: 1

Tamanho em disco: 1,0MB

Com este recurso, foi possível extrair as LineString e os Points presentes do arquivo CAD original. Foi interpretada uma feição de Polygon, com tamanho bastante reduzido, que corresponde a um elemento da legenda. O *layer* de GeometryCollection interpretou 1 LineString.

É interessante notar que os metadados geométricos acima, são os mesmos apresentados quando realizamos a *abertura* do arquivo CAD 3313-164 no QGIS 2.0.1, com exceção do GeometryCollection que foi indicado pelo sistema, mas não recuperado graficamente no SIG.

As coordenadas UTM foram repassadas corretamente ao formato de saída.

As isolinhas perderam seu valor de altitude.

4.1.2.1) Conversão do arquivo para ESRI Shapefile via OGR no QGIS 2.0.1

Para efeito de comparação, faremos a *conversão* do arquivo 3313-164 carregado na versão mais nova do QGIS, utilizando a *implementação gráfica* da OGR no SIG:

Figura 41. Conversão do arquivo 3313-164 de DXF para ESRI Shapefile, utilizando a GDAL/OGR implementada graficamente no QGIS.

Os metadados após a conversão são:

Polygon: 1
 LineString: 375
 Point: 467
 GeometryCollection: 0
 Tamanho em disco: 1,5MB

O arquivo ESRI Shapefile reconheceu as geometrias de LineString, Point (com seus *labels* respectivos) e Polygon (elemento da legenda). Apesar do Polygon ter sido recuperado, a maior parte dos elementos geométricos da legenda foram perdidos (o *layer* de GeometryCollections apresentou falha na conversão e ficou vazio, sem conteúdo).

Os metadados geométricos acima são coincidentes com aqueles gerados na abertura da base CAD 3313-164 no QGIS 2.0.1.

Todos os *layers* estão geoposicionados segundo o EPSG 29193 - o mesmo da base CAD.

A exportação fez com que as geometrias 3D do CAD, perdessem seus atributos de altitude.

4.1.2.2) Conversão de AutoCAD DXF para ESRI Shapefile pelo recurso de *'explodecollections'*

Com o emprego deste recurso, obtivemos o melhor resultado para a interpretação do formato DXF pela OGR. Desta forma, é possível que sua utilização em conjunto com a conversão para o formato ESRI Shapefile, produza também bons resultados.

O comando é:

```
ogr2ogr -f "ESRI Shapefile" -explodecollections 3313-164.shp 3313-164.DXF
```

...e o resultado segue abaixo:

```
Warning 6: Normalized/launched field name: 'ExtendedEntity' to 'ExtendedEn'  
Warning 6: Normalized/launched field name: 'EntityHandle' to 'EntityHand'  
ERROR 1: Attempt to write non-point (LINESTRING) geometry to point shapefile.  
ERROR 1: Terminating translation prematurely after failed  
translation of layer entities (use -skipfailures to skip errors)
```

A biblioteca gerou o arquivo *shapefile*, mas vazio, sem conteúdo. Isto mostra que o recurso *'explodecollections'* foi melhor aplicável no formato DXF, não fazendo efeitos (positivos) no formato analisado.

4.1.2.3) Conversão dos arquivos DXF (originados do OGR *explodecollections* e explodidos no DraftSight) em ESRI Shapefile

Uma vez que a OGR falha em explodir os blocos CAD em sua conversão para o ESRI Shapefile, é possível que o arquivo CAD com blocos já eliminados, seja reconhecido com mais perfeição. Para isto, rodamos os comandos nas duas versões do DXF que testamos até o momento.

a) Para o arquivo originado do OGR *'explodecollections'* rodamos o seguinte comando

```
ogr2ogr -f "ESRI Shapefile" geo.shp geo.dxf
```

...e obtivemos o resultado a seguir:

```
Warning 6: Normalized/laundered field name: 'ExtendedEntity' to 'ExtendedEn'
Warning 6: Normalized/laundered field name: 'EntityHandle' to 'EntityHand'
ERROR 1: Attempt to write non-point (LINESTRING) geometry to point shapefile.
ERROR 1: Terminating translation prematurely after failed
translation of layer entities (use -skipfailures to skip errors)
```

b) Para o arquivo 3313-164_copia_R2000-2002.dxf (blocos explodidos no DraftSight):

```
ogr2ogr -f "ESRI Shapefile" geoexplode.shp 3313-164_copia_R2000-2002.dxf
```

```
Warning 6: Normalized/laundered field name: 'ExtendedEntity' to 'ExtendedEn'
Warning 6: Normalized/laundered field name: 'EntityHandle' to 'EntityHand'
ERROR 1: Attempt to write non-point (LINESTRING) geometry to point shapefile.
ERROR 1: Terminating translation prematurely after failed
translation of layer entities (use -skipfailures to skip errors)
```

Em ambas as conversões, foi gerada a estrutura do arquivo *shapefile*, mas vazio, sem conteúdo.

4.1.2.4) Conclusão da conversão de AutoCAD DXF para ESRI Shapefile

O aplicativo **ogr2ogr** falhou em converter entre os formatos, gerando um arquivo ESRI Shapefile vazio, sem conteúdo. Contudo, sua utilização simultânea ao recurso de separação de geometrias '**ogr_geometry**' foi bem sucedida, indicando que a restrição geométrica do formato SHP (um único tipo de geometria suportada por *layer*), ocasione problemas na conversão de múltiplas geometrias extraídas de uma única camada (como é o caso do *layer* DXF 'entities').

A utilização da GDAL/OGR implementada no QGIS 2.0.1, apresentou resultado inferior ao aplicativo **ogr2ogr** com o recurso '**ogr_geometry**' ativado, pois a implementação gráfica falhou em extrair a LineString do *layer* GeometryCollection.

Nas duas conversões, as coordenadas UTM foram transmitidas com sucesso aos formatos de saída.

O recurso de '**explodecollections**' não foi eficiente na interpretação do *shapefile* pela biblioteca, pelo contrário, o arquivo convertido ficou vazio, sem conteúdo.

Concluimos que a OGR consegue converter arquivos DXF para ESRI Shapefile, contudo, o aplicativo **ogr2ogr** sem o recurso '**ogr_geometry**' não é suficiente para o processo.

4.1.3) Conversão de AutoCAD DXF para MapInfo MIF/MID

O MapInfo, um dos SIG proprietários mais importantes da atualidade, emprega o módulo *Universal Translator*³⁸ para a importação e exportação de dados no sistema.

Avaliaremos a performance da OGR na conversão de dados CAD para o formato padrão de intercâmbio de dados do SIG.

³⁸ Tecnologia FME, licenciada da empresa Safe Software Inc.

O comando de conversão para o MIF/MID é³⁹:

```
ogr2ogr -f "MapInfo File" 3313-164.mif 3313-164.DXF
```

Resultado:

a) Pelo OpenJUMP:

Este sistema não suporta o formato.

b) Pelo CAD DraftSight:

Este sistema não suporta o formato.

c) Pela OGR:

```
ogrinfo -so 3313-164.mif  
ogrinfo -so 3313-164.mif  
Had to open data source read-only.  
INFO: Open of `3313-164.mif'  
      using driver `MapInfo File' successful.
```

As informações acima mostram que o *driver* da biblioteca consegue interpretar o novo arquivo convertido.

d) Pelo QGIS 2.8.1

³⁹ Para este teste, foi utilizado o QGIS versão 2.8.1 -Wien, compilado sobre a GDAL/OGR 1.11.1.

Figura 42. Visualização no QGIS da importação do arquivo 3313-164, convertido para o formato MIF/MID pelo aplicativo **ogr2ogr**.

Os metadados do resultado da conversão são:

LineString: 4754
 Point: 467
 Polygon: 16
 Tamanho em disco: 1,7MB

Notar que o número de Points permaneceu o mesmo, em relação ao arquivo ESRI Shapefile convertido do DXF e importado no QGIS 2.0.1.

As LineStrings, por sua vez, foram multiplicadas provavelmente porque este formato quebrou em mais segmentos (de reta) as geometrias lineares da legenda.

Existiram perdas nos elementos da legenda e problemas com caracteres especiais dos textos no mapa (como 'é', '^' e 'ç', por exemplo).

As curvas de nível perderam o valor de altitude após a conversão.

Para tentar solucionar os problemas apresentados nos textos e na legenda, aplicamos o recurso de *'explodecollections'* na conversão para o MIF-MID (comando: **ogr2ogr -f "MapInfo File" -explodecollections 3313-164_2.mif 3313-164.DXF**) e o resultado foi o mesmo da conversão sem o recurso, sem efeitos positivos ou negativos nos textos e na legenda.

4.1.3.1) Conclusão da conversão de AutoCAD DXF para MapInfo MIF/MID

Comparado ao ESRI Shapefile, o MIF/MID não aplica restrições geométricas aos *layers*, o que significa que o formato consegue suportar as múltiplas geometrias do formato DXF (condensadas sob o *layer* 'entities'), sem a necessidade do recurso de separação de geometrias (**ogr_geometry**).

É possível que, sendo ambos DXF e MIF/MID baseados em texto, o processo de conversão pela biblioteca seja simplificado, bastando mapear os esquemas em ASCII nos dois formatos. Contudo, isto não garantiu uma conversão livre de falhas, mesmo que pequenas.

O QGIS reconheceu corretamente as coordenadas da projeção UTM.

Uma vez que o MIF/MID não suporta geometrias 3D, as isolinhas perderam o atributo de elevação. Esta restrição se deve ao modelo conceitual adotado na estruturação das geometrias empregadas no formato.

Concluimos que a OGR pode ser utilizada para a conversão de dados em DXF para o MIF/MID, pois o conteúdo geométrico, textos e coordenadas do mapa foram mantidos. Os problemas da conversão estão relacionados aos caracteres especiais dos textos, dentro e fora da legenda.

4.1.4) Conversão de AutoCAD DXF para Google KML

A possibilidade de importar dados para o Google Earth e utilizar sua coleção de imagens satelitais de alta resolução espacial, bem como modelos 3D de superfície (edifícios, árvores, etc) e terreno (MNT), pode ser de grande importância nos campos profissional e de pesquisa em Geografia e Cartografia. Adicionalmente, aproveitar a capacidade dos SIG em processar dados geoespaciais para posterior carregamento no Google Earth, fazem da conversão ao KML um teste significativo para a OGR.

A conversão do formato DXF para o KML é feita pelo seguinte comando:

```
ogr2ogr -f "KML" 3313-164.kml 3313-164.DXF
```

ERROR 1: Latitude 7393284.126328 is invalid. Valid range is [-90,90]. This warning will not be issued any more

Warning 1: Longitude 324527.755582 has been modified to fit into range [-180,180]. This warning will not be issued any more

O 'ERROR 1' foi informado porque o formato KML não possui suporte nativo para coordenadas em metros (no caso, o arquivo DXF está em coordenadas projetadas UTM). O formato reconhece apenas o georreferenciamento representado pelo código EPSG: 4326.

A abertura deste arquivo convertido pela biblioteca nos SIG, é ilustrada na figura 35:

a) Pelo OpenJUMP:

Figura 43. Visualização do arquivo KML no openJUMP, após conversão do formato DXF. O driver KML do OpenJUMP não conseguiu interpretar corretamente o mapa.

Metadados:

Nome do layer: 3313-164

Número de feições: 844

Número de pontos: 13702

Notar que o QGIS interpretou melhor as geometrias em relação ao OpenJUMP, porém não foi suficiente para a correta apresentação do mapa.

Os elementos geométricos, além da distorção, foram embaralhados em relação ao mapa original. Os pontos foram convertidos, mas os textos correspondentes apresentaram problemas de codificação (caracteres especiais).

É provável que a conversão tenha falhado devido à questão do referencial geodésico, uma vez que o formato KML é restritivo neste aspecto. Por mais que a biblioteca tenha atribuído, automaticamente, novos parâmetros de coordenadas, é possível que recurso não surtiu efeitos (positivos) no arquivo de saída.

Para testar essa hipótese, utilizaremos a biblioteca para reprojetar⁴¹ o arquivo para o EPSG: 4326 durante a conversão, indicando que o referencial de origem da base CAD é o EPSG: 29193. Empregaremos, adicionalmente, o recurso ‘*explodecollections*’ na conversão. O comando é:

```
ogr2ogr -f "KML" -explodecollections 3313-164.kml 3313-164.DXF -t_srs  
EPSG:4326 -s_srs EPSG:29193
```

A figura 45 ilustra o arquivo KML reprojetado após sua conversão do CAD, com os metadados indicados:

Metadados:

Point: 467

LineString: 4754

Polygon: 16

Tamanho em disco: 3,9MB

⁴¹ Conforme estabelece o IBGE: “ [...] com o término do período de transição para SIRGAS2000, em 25 de fevereiro de 2015, os resultados de qualquer trabalho devem ser referidos exclusivamente a este sistema. Por outro lado [...] o WGS 84 atualmente pode ser considerado, para fins práticos, coincidente com o SIRGAS2000. Portanto, basta o usuário referir seus resultados ao SIRGAS2000 que, automaticamente, produzirá resultados em WGS 84 (e vice-versa)”. Adicionalmente, a Resolução do Presidente do IBGE nº 1, de 25/02/2005, apresenta os parâmetros de transformação entre o SAD69 e o SIRGAS2000. Fonte: <<http://www.ibge.gov.br/home/geociencias/geodesia/pmrg/faq.shtm>>. Acesso em 24 jun. 2015.

Figura 45. Conversão do CAD para KML, com os parâmetros da reprojeção EPSG estabelecidos.

Podemos perceber que a configuração dos parâmetros da reprojeção, evitou a distorção das diferentes geometrias no formato de saída.

As LineStrings continuaram sem seus valores de altitude; os Points, por sua vez, apresentaram valores dos textos (com alguns problemas no gerenciamento de caracteres especiais como cedilha, acentos agudos e circunflexos).

Quando recuperado no Google Earth, este aplicativo reconheceu as coordenadas espaciais reprojatadas e parte dos textos (condensados nos ícones do aplicativo), como mostra a figura 46:

Figura 46. Dados KML carregados no Google Earth, após a conversão do formato DXF pela OGR.

Os desvios dos vetores com relação à imagem, podem se dar por dois fatores relacionados:

- 1) pela ortorretificação empregada pelo Google, e;
- 1) pela diferença entre as coordenadas da base original de restituição (ortofotos) passadas aos vetores, com aquelas das imagens satelitais do Google; ou seja, diferenças de geoposicionamento entre as bases.

Finalmente, para avaliarmos se a ausência de blocos no arquivo CAD original facilitaria a conversão para o KML, explodimos os blocos e reescrevemos o 3313-164 para a versão 2000-2002 do DXF.

Após a conversão, a única diferença perceptível foi com relação à codificação, já que o KML proveniente do DXF sem blocos, não apresentou problemas com os caracteres especiais dos textos.

4.1.4.1) Conclusão da conversão de AutoCAD DXF para KML

A conversão só foi bem sucedida após a reprojeção do arquivo durante o processo, pois o formato KML possui restrição geodésica suportando apenas o EPSG: 4326.

Parte do texto (como valores dos pontos cotados e nomes dos edifícios, por exemplo), foram recuperados no Google Earth e no QGIS.

Concluimos que, apesar de problemas com caracteres especiais - resolvido com a eliminação dos blocos no arquivo CAD original -, a OGR pode ser utilizada para a conversão de dados DXF para o KML, pois os elementos geométricos e as coordenadas espaciais, foram bem repassadas ao formato de saída.

4.1.5) Conversão de AutoCAD DXF para GML

Apesar de ambos os formatos serem baseados em textos estruturados, a OGR deve ser capaz de mapear os esquemas em ASCII para o XML (base do GML) para que a conversão seja bem-sucedida.

Segundo a documentação técnica, a OGR possui suporte limitado ao formato, em leitura e escrita.

O comando para a conversão é:

```
ogr2ogr -f "GML" 3313-164.gml 3313-164.DXF
```

Resultado:

a) Carregamento no OpenJUMP:

Apesar de oferecer suporte ao GML (versão 2.0), este SIG não conseguiu importar o arquivo, muito provavelmente porque a versão do GML produzida pela OGR foi a 2.1.2.

b) Carregamento no QGIS 2.8⁴²:

⁴² Neste teste, a versão da GDAL/OGR utilizada é a 1.11.1 e a do QGIS é a 2.8.0 - Wien (versão de desenvolvimento) e a final 2.8.1 - Wien (também compilada sobre a GDAL/OGR 1.11.1).

Este SIG carrega os *layers* ‘entities’, com os seguintes parâmetros:

Nome da Camada	Número de Feições	Tipo de Geometria
entities	467	Point25D
entities	375	LineString25D
entities	1	Polygon25D
entities	1	GeometryCollection25D
Tamanho em disco: 2,5MB		

A notação “25D” após as geometrias, indica que a GDAL/OGR implementada graficamente no QGIS, força as geometrias a serem carregadas com seus valores 3D (se disponíveis).

O mapa aberto no SIG, é ilustrado na figura 47:

Figura 47. Ilustração do arquivo GML carregado no QGIS, após a conversão do formato DXF.

As coordenadas UTM foram repassadas corretamente ao formato de saída.

A legenda perdeu seus elementos geométricos e apresentou problemas com relação aos caracteres especiais (‘é’ e ‘ç’).

Para tentar solucionar os problemas na legenda, utilizamos o comando “*explodecollections*” (**ogr2ogr -f "DXF" -explodecollections geo.dxf 3313-164.DXF**) no arquivo CAD, afim de separar as geometrias condensadas no *layer* GeometryCollection. O arquivo resultante apresentou os seguintes metadados:

Nome da Camada	Número de Feições	Tipo de Geometria
entities	467	Point25D
entities	12227	LineString25D
entities	19	Polygon25D
Tamanho em disco: 7,0MB		

Para avaliarmos a resposta gráfica do novo arquivo GML, abrimo-lo no QGIS conforme ilustra a figura 49:

Figura 49. Arquivo GML carregado no QGIS, após a explosão dos blocos no sistema CAD e posterior conversão pela OGR.

Como podemos perceber, a explosão dos blocos no sistema CAD permitiu uma excelente recuperação dos itens antes suprimidos na legenda, no formato GML.

4.1.5.1) Conclusão da Conversão de AutoCAD DXF para GML

A presença do elemento “bloco” no arquivo CAD, prejudicou a conversão para o formato GML.

O recurso “*explodecollections*” da OGR não foi suficiente para eliminar completamente os blocos no formato DXF.

O melhor resultado com o GML, inclusive mantendo símbolos (ícones) e caracteres especiais da legenda ('ç', '^' e '~', por exemplo), deu-se após a eliminação dos blocos no sistema DraftSight, indicando mais uma vez que a OGR possui suporte parcial ao elemento bloco na estrutura CAD.

É interessante notar que, após a explosão dos blocos no CAD e a subsequente conversão para o GML, as LineStrings foram segmentadas em seus vértices.

4.1.6) Conversão de AutoCAD DXF para Microstation DGN

Apesar de popular no universo SIG, o DGN é um formato vetorial CAD. Instituições como o Exército Brasileiro e IBGE, por exemplos, costumam gerenciar dados neste formato.

A conversão de DXF para DGN deveria ser, teoricamente, simples e bem-sucedida pois ambos os formatos são estruturados como desenho CAD. A documentação técnica da OGR, entretanto, afirma que o *driver* possui suporte parcial ao formato de saída.

Analisaremos, em seguida, a performance da OGR na conversão entre os dois formatos CAD. O comando é:

```
ogr2ogr -f "DGN" 3313-164.dgn 3313-164.DXF
```

Resultado:

a) Pelo OpenJUMP:

O sistema não suporta este formato.

b) Pelo QGIS 2.8.1:

Figura 50. Layer do arquivo 3313-164.dgn aberto no QGIS.

Metadados:

Metadados:

LineString: 4754

Point: 467

Polygon: 16

Tamanho em disco: 602,1KB

Alguns Itens textuais da legenda apresentaram problemas de codificação (caracteres especiais), enquanto outros não foram recuperados graficamente no SIG.

As coordenadas UTM foram transmitidas corretamente ao formato de saída.

As geometrias também foram apresentadas de maneira correta no QGIS.

Novamente, aplicamos o recurso de *'explodecollections'* na conversão, e o resultado foi o mesmo daquele sem os blocos eliminados.

4.1.6.1) Conclusão da conversão de DXF para DGN

Apesar de ambos estarem estruturados como formatos CAD, o reconhecimento da OGR não foi perfeito com relação ao DGN. A própria biblioteca indica a falta de suporte completo ao formato.

Contudo, com exceção da codificação dos textos (caracteres especiais) e omissão de alguns elementos textuais da legenda, a OGR foi bem sucedida em transmitir corretamente as coordenadas espaciais e as geometrias do formato de entrada. Na verdade, o mapa de saída no formato DGN ficou muito próximo de outros testados nesta pesquisa.

Concluimos que a OGR pode ser utilizada para conversões de DXF para DGN, porém os elementos textuais dos mapas ficarão prejudicados, sobretudo aqueles vinculados aos blocos do CAD DXF.

4.2 Leitura e Escrita: ESRI Shape para AutoCAD DXF

Até agora, os testes realizados envolveram a conversão de dados na estrutura CAD para a SIG, com o objetivo de avaliar a performance da biblioteca OGR no aspecto da interoperabilidade sintática.

No teste a seguir, avançaremos e avaliaremos o processo inverso, qual seja, o de converter por meio da OGR, dados formatados na estrutura SIG para a CAD. O formato escolhido será o ESRI Shapefile, por ser interoperável e o padrão dos SIG aqui utilizados.

4.2.1) Carregamento do arquivo ESRI Shape do MDC 1:1.000 em SIG

Como o ESRI Shapefile é bem suportado nos SIG, faremos seu carregamento somente no QGIS. Os metadados para este arquivo são:

Geral

Tipo de armazenamento dessa camada

ESRI Shapefile

Descrição deste provedor

OGR data provider (compiled against GDAL/OGR library version 1.10.0, running against GDAL/OGR library version 1.10.0)

Tipo de geometria das feições nessa camada

Polygon

O número de feições nesta camada

12904

Sistema de unidades de referência espacial na camada

xMin,yMin 321025.68, 7391301.99 :xMax, yMax 326045.45, 7395929.83

Sistema de referência espacial da camada

+proj=utm +zone=23 +south +ellps=aust_SA +towgs84=-57,1,-41,0,0,0,0 +units=m
+no_defs

Mesmo geradas por meio de *planar enforcement*, utilizamos o recurso de verificação de topologia do QGIS, para a identificação de erros quanto a polígonos duplicados e/ou com *overlaps*. Não foram encontrados erros.

Figura 51. Ilustração do arquivo *shapefile*, objeto dos próximos testes, aberto no QGIS.

Esta base apresenta, entre outros atributos, as alturas das edificações na área do setor, armazenadas no campo ALTURA_EDI do arquivo DBF.

Entre os usos possíveis, podemos gerar a extrusão⁴³ de todos os edifícios mapeados por meio desse campo, como mostra a figura 52, o que indica que sua preservação nos processos de conversão, é fundamental:

Figura 52. Modelagem 3D do arquivo *shapefile* no SIG GRASS.

Convém ressaltar que o GRASS armazena os sólidos (no exemplo, prismas) em formato próprio (baseado em ASCII) após a importação do *shapefile* e processamento do recurso *extrude*, **não** no formato ESRI Shapefile com o tipo geométrico *Multipatch*.

A extrusão do *shapefile* de 3,8MB referente ao setor 82 no GRASS, resultou em arquivos no disco totalizando 34,3MB.

Recentemente, o OGC definiu o formato CityGML (baseado em XML), versão 2, como o padrão para representação tridimensional de *idades virtuais* (representação de estádios, edifícios, pontes, túneis, vegetação, corpos d'água, modelos numéricos de terreno, etc). Mais detalhes podem ser vistos em: <<http://www.opengeospatial.org/standards/citygml>>. Acesso em 12 mar. 2015.

⁴³ A extrusão permite gerar sólidos a partir de geometrias como Polygon e LinearRing, aumentando consideravelmente o número de vértices e arestas (segmentos) em relação às geometrias 2D, dado que mais uma dimensão (Z) é utilizada na renderização dos objetos.

A partir da versão 1.8, a GDAL/OGR tem suporte parcial de leitura em arquivos CityGML.

4.2.2) Conversão de ESRI Shapefile para AutoCAD DXF

Uma vez que o atributo de altura das edificações está armazenado no banco de dados do arquivo *shapefile*, e não como uma propriedade das geometrias (Polygons) do *layer*, a conversão pela biblioteca resultará na perda deste valor em CAD (pois o formato DXF não gerencia estruturas de banco de dados).

A GDAL/OGR possui um recurso (**-zfield**) que permite extrair os valores Z de atributos SIG e transferí-los às geometrias com suporte 3D. Isto significa que as alturas dos edifícios (armazenadas no campo ALTURA_EDI do banco de dados), podem ser especificadas como uma propriedade das geometrias no formato CAD. O comando completo é:

```
ogr2ogr -f "DXF"setor82.dxf setor82.shp-zfield ALTURA_EDI
```

Na conversão, as geometrias e as coordenadas UTM foram mantidas porém os valores das alturas não foram atribuídos aos polígonos no formato de saída, conforme indicam os parâmetros da feição, abaixo:

```
OGRFeature(entities):12901
  Layer (String) = 0
  SubClasses (String) = AcDbEntity:AcDbHatch
  ExtendedEntity (String) = (null)
  Linetype (String) = (null)
  EntityHandle (String) = 23265
  Text (String) = SOLID
  Style = BRUSH(fc:#000000)

POLYGON ((322011.010087009985 7394222.280023399740458,322018.78
7394224.37,322021.19 7394215.41,322015.850924252998084
7394213.996715240180492,322017.239735521026887
7394209.060939949937165,322014.679200745013077
7394208.37302939966321,322011.010087009985 7394222.280023399740458))
```


Uma hipótese a este problema é que o recurso de atribuição de altura às geometrias da GDAL/OGR, foi criado tendo em vista isolinhas altimétricas, ou seja, geometrias lineares. E se os desenvolvedores não previram a utilização de dados de elevação em polígonos, somente em linhas?

Para testar nossa hipótese, utilizamos o QGIS e convertimos todos os Polygons do arquivo em LinearRings (a GDAL/OGR não possui este recurso). Com isto, obtivemos estruturas lineares, porém fechadas. Por fim, refizemos a operação na biblioteca e os dados de altura foram gravados com sucesso no formato de saída, como indicam os parâmetros em negrito (indicando a altura do edifício) e a figura 53:

```
OGRFeature(entities):12901
```

```
Layer (String) = 0
```

```
SubClasses (String) = AcDbEntity:AcDbPolyline
```

```
ExtendedEntity (String) = (null)
```

```
Linetype (String) = (null)
```

```
EntityHandle (String) = 23265
```

```
Text (String) = (null)
```

```
Style = PEN(c:#000000)
```

```
LINestring (322160.467535071016755 7392897.147753830067813
```

```
3.783,322165.116350180993322 7392900.581856509670615
```

```
3.783,322168.938362782006152 7392895.725182440131903
```

```
3.783,322159.907824478985276 7392889.059186319820583
```

```
3.783,322154.373568381997757 7392884.97401128988713
```

```
3.783,322153.235877045022789 7392886.514127340167761
```

```
3.783,322154.48138289898634 7392887.43418863043189
```

```
3.783,322154.19766249897657 7392887.818266740068793
```

```
3.783,322156.267361558973789 7392889.347163680009544
```

```
3.783,322154.799283598025795 7392891.334530870430171
```

```
3.783,322157.564850841008592 7392893.377469070255756
```

```
3.783,322157.212922472972423 7392893.853881710208952
```

```
3.783,322160.892722845019307 7392896.572168430313468
```

```
3.783,322160.467535071016755 7392897.147753830067813 3.783)
```

Figura 53. Resultado da conversão dos edifícios (como LinearRings) ao formato DXF, por meio do recurso **-zfield**.

Agora, as feições apresentam as informações de altura, com o custo de terem suas geometrias transformadas. Em alguns sistemas CAD, estas geometrias podem ser extrudadas e visualizadas em 3D, assim como apresentadas no GRASS. O DraftSight não possui este recurso.

É importante notar que após a conversão de Polygon para LinearRing e a consequente reescrita do arquivo para DXF, a biblioteca alterou a numeração dos indentificadores das feições (OGRFeatures), o que explica a alteração das coordenadas UTM ao se comparar os dois parâmetros geométricos anteriores.

4.3 Leitura e Escrita: ArcInfo Grid para GeoTIFF

Avaliaremos no próximo teste, o comportamento da GDAL na conversão do formato ArcInfo Grid para o GeoTIFF. Enquanto o primeiro é apenas parcialmente suportado (somente leitura) pela GDAL, o segundo possui suporte completo (leitura e escrita).

A conversão pode ser necessária caso o SIG utilizado (como exemplo, o Global Mapper) não suporte o formato proprietário da ESRI.

Para o teste, converteremos todo o MNT Ikonos da Região Metropolitana de São Paulo.

A figura 54 ilustra o MNT da RMSP em ArcInfo Grid carregado no QGIS:

Figura 54. Notar que o MNT foi corretamente carregado no SIG, o que significa que a GDAL interpretou corretamente o formato (em modo de leitura).

Foram aplicadas falsas cores para as classificações dos *pixels*.

O arquivo ArcInfo Grid carregado no QGIS, possui os seguintes metadados:

```
Driver
GDAL provider
AIG
Arc/Info Binary Grid
Descrição do registro
/home/linux/3313-164/mdt
Banda 1
STATISTICS_MAXIMUM=1255
STATISTICS_MEAN=786,1260115682
STATISTICS_MINIMUM=250
STATISTICS_STDDEV=95,660552329338
Dimensões
X: 31574 Y: 20511 Bandas: 1
Origem
273226,7.43772e+06
Tamanho do Pixel
5,-5
Sem valor de dados
-32768
Tipo de Dados
Int16 - Inteiro de 16 bits com sinal
Visão geral das pirâmides
```

```

Sistema de Referência Espacial da Camada
+proj=utm +zone=23 +south +datum=WGS84 +units=m +no_defs
Extensão da camada (fonte de projeção original da camada)
273225.6716980165801942,7335160.2792629878968000 :
431095.6716980165801942,7437715.2792629878968000
Banda
Banda 1
No da Banda
1
Sem Informações
Nenhuma informação coletada até o momento

```

Notar que os metadados listam a resolução espacial do *pixel* em 25 metros (5x5m); a referência espacial como UTM, WGS84, fuso 23, sul (código EPSG 32723); o tipo de dado como 16-bit (resolução radiométrica) e a extensão da camada em coordenadas UTM.

Em disco, o MNT composto por 9 arquivos no formato original (ArcInfo Grid) ocupa aproximadamente 300MB.

O comando de conversão para o GeoTIFF, utilizando a GDAL é:

```

gdal_translate -of GTiff w001001.adf rmsp_mnt.tif
Warning 6: A dataset opened by GDALOpenShared should have the same
filename (w001001.adf) and description (.)
Input file size is 31574, 20511
0...10...20...30...40...50...60...70...80...90...100 - done.

```

O resultado da conversão é ilustrado pela figura 55:

Figura 55. GeoTIFF gerado pela GDAL e carregado no QGIS.

Foram aplicadas falsas cores para as classificações dos *pixels*.

O novo arquivo GeoTIFF possui os seguintes metadados:

```
Driver
GDAL provider
GTiff
GeoTIFF
Descrição do registro
/home/linux/3313-164/mdt/rmsp_mnt.tif
AREA_OR_POINT=Area
Banda 1
STATISTICS_MAXIMUM=1368
STATISTICS_MEAN=784,7868421674
STATISTICS_MINIMUM=20
STATISTICS_STDDEV=119,96452663134
Dimensões
X: 31574 Y: 20511 Bandas: 1
Origem
273226,7.43772e+06
Tamanho do Pixel
5,-5
Sem valor de dados
-32768
Tipo de Dados
Int16 - Inteiro de 16 bits com sinal
Visão geral das pirâmides
Sistema de Referência Espacial da Camada
+proj=utm +zone=23 +south +datum=WGS84 +units=m +no_defs
```

```

Extensão da camada (fonte de projeção original da camada)
273225.6716980165801942,7335160.2792629878968000 :
431095.6716980165801942,7437715.2792629878968000
Banda
Banda 1
No da Banda
1
Sem Informações
Nenhuma informação coletada até o momento

```

O novo MNT em GeoTIFF ocupa 1,2GB em disco, quatro vezes o tamanho do *raster* original no formato ArcInfo Grid. Uma forma de diminuir o tamanho do arquivo em disco, é utilizar um método de compressão com a conversão. O recurso **-co COMPRESS=ZIP**, adicionado ao final do comando da GDAL, aplica a compressão ZIP⁴⁴ ao formato TIFF. Em nosso teste, o arquivo comprimido com este recurso, ocupou 178,9MB em disco - um pouco mais da metade do arquivo ArcInfo Grid original.

É interessante notar que os principais metadados como resolução espacial, referência geodésica e resolução radiométrica foram preservados.

As diferenças entre as estatísticas para a Banda 1 dos formatos de entrada e saída, dizem respeito aos valores de *pixel* (altitude) selecionados para a amostragem e classificação da imagem (falsa cor).

Adicionalmente, é possível utilizar o aplicativo **gdalwarp** para fazer a reprojeção do arquivo matricial gerado (EPSG:32723), e torná-lo, por exemplo, compatível espacialmente com o MDC e Ortofotos utilizados nesta pesquisa (EPSG:29193). O comando e seu resultado no console, são apresentados a seguir:

```
gdalwarp -t_srs EPSG:29193 rmsp_mnt.tif rmsp_mnt_epsg29193.tif
```

```

Creating output file that is 31574P x 20511L.
Processing input file rmsp_mnt.tif.
Using internal nodata values (e.g. -32768) for image rmsp_mnt.tif.
Copying nodata values from source rmsp_mnt.tif to destination
rmsp_mnt_epsg29193.tif.
0...10...20...30...40...50...60...70...80...90...100 - done.

```

44 Método de compressão utilizado em arquivos de texto e imagem. Mais detalhes em: <<https://www.pkware.com/support/zip-app-note/>>. Acesso em 12 mar. 2015.

O parâmetro “-t_srs” na sequência do comando, traduz-se como “*target spatial reference system*”, indicando que é necessário especificar o código EPSG para a transformação entre os referenciais geodésicos.

O novo arquivo reprojetoado, também em GeoTIFF, possui os seguintes metadados:

```
Driver
GDAL provider
GTiff
GeoTIFF
Descrição do registro
/home/linux/3313-164/mdt/rmsp_mnt_epsg29193.tif
AREA_OR_POINT=Area
Banda 1
STATISTICS_MAXIMUM=1368
STATISTICS_MEAN=453,22961842726
STATISTICS_MINIMUM=0
STATISTICS_STDDEV=398,22470688341
Dimensões
X: 31574 Y: 20511 Bandas: 1
Origem
273265,7.43776e+06
Tamanho do Pixel
5,-5
Sem valor de dados
-32768
Tipo de Dados
Int16 - Inteiro de 16 bits com sinal
Visão geral das pirâmides
Sistema de Referência Espacial da Camada
+proj=utm +zone=23 +south +ellps=aust_SA +towgs84=-57,1,-41,0,0,0,0 +units=m +no_defs
Extensão da camada (fonte de projeção original da camada)
273265.1759273274801672,7335204.8084075888618827 :
431135.2194955531158485,7437759.8367102388292551
Banda
Banda 1
No da Banda
1
Sem Informações
Nenhuma informação coletada até o momento
```

A reprojeção alterou a referência espacial do arquivo, enquanto manteve os parâmetros de resolução espacial e radiométrica.

Uma vez carregado no QGIS, o sistema reconheceu automaticamente o georreferenciamento do novo arquivo.

4.3.1) Conclusão da conversão do MNT em ArcInfo Grid para o GeoTIFF

A GDAL se mostrou muito eficiente na conversão do formato proprietário ArcInfo Grid (o qual é suportado apenas parcialmente, em modo de leitura) para o GeoTIFF (interoperável e que está se tornando padrão na área).

Não existiram perdas de dados e informações na conversão entre os formatos.

Como mencionado, essa conversão se torna interessante para a migração de dados aos SIG que não possuem suporte ao formato original.

CAPÍTULO 5 - INTEGRAÇÃO DA GDAL/OGR COM SISTEMA GERENCIADOR DE BANCO DE DADOS ESPACIAIS

Nosso último teste consiste em avaliar a integração da biblioteca GDAL/OGR com o sistema Spatialite⁴⁵, por meio do armazenamento de dados suportados pela biblioteca no SGBD Espacial.

Spatialite é uma extensão espacial ao banco de dados *open source* SQLite⁴⁶, transformando-o em um Sistema Gerenciador de Banco de Dados (SGBD) Espaciais, ou seja, em um sistema com recursos típicos de SIG como consultas espaciais, análises métricas e topológicas, classificadores temáticos entre outros. O Spatialite é, junto ao ESRI Shapefile, o *formato* padrão do QGIS em sua nova versão (a partir da 2). O formato também é suportado por sistemas proprietários como ArcGIS, MapInfo e Global Mapper, por exemplo.

Existem vantagens na adoção de um formato de dados geoespaciais baseado em SGBD, como propõe Davis:

“A partir do momento em que um SGBD espacial independente do GIS torna-se disponível [...], uma estruturação completamente nova do ambiente de geoprocessamento pode ser implementada. Em primeiro lugar, como os dados geográficos passam a ser gerenciados exclusivamente por um SGBD padronizado, **o formato de armazenamento específico do GIS perde importância. Também os formatos de intercâmbio perdem importância, uma vez que seria possível para GIS diferentes compartilhar os mesmos dados, no mesmo SGBD – e, se necessário, armazená-los localmente no formato julgado mais conveniente.**” (DAVIS, 2003, pp.26-37, grifo nosso).

45 Informações adicionais daquelas mencionadas aqui, consultar: <<https://www.gaia-gis.it/fossil/libspatialite/index>>. Acesso em 02 jul. 2014.

46 SQLite é um dos bancos de dados mais utilizados no planeta, apoiando diversos aplicativos como o sistema operacional Mac OSX, o navegador Firefox, o comunicador Skype, etc. Diferentemente de outros SGBD, o SQLite não está estruturado na arquitetura *cliente - servidor*, o que simplifica o processo de acesso e gerenciamento do sistema, uma vez que os dados, seus processos de gravação e armazenamento no SGBD, são armazenados no mesmo arquivo em disco, que funciona de forma interoperável em distintas plataformas (Windows, Linux, OSX, etc), tanto em 32 quanto em 64-Bit. Mais informações em: <<http://www.sqlite.org/index.html>>. Acesso em 02 jul 2014.

Isto é verdadeiro para dados que serão gerados *a posteriori* ou que foram previamente convertidos para o *formato* utilizado no SGBD Espacial. Mas e os demais dados que não puderam ser convertidos e que estão em formatos não suportados pelos SGBD Espaciais utilizados? A GDAL/OGR seria suficiente para o processo de gravação de dados vetoriais e matriciais no SGBD Espacial, eliminando a necessidade de formatos de intercâmbio de dados?

Para responder a essas questões, testaremos recursos de conversão de dados da OGR em conjunto com o Spatialite, avaliando se dados vetoriais suportados pela biblioteca podem ser armazenados no SGBD Espacial e acessados diretamente pelos SIG (Global Mapper e QGIS). O arquivo escolhido para o armazenamento no SGBD será o mesmo dos testes anteriores: O MDC 3313-164.DXF.

Para os *rasters*, testaremos os recursos de criação e gerenciamento de dados das bibliotecas GDAL e Rasterlite⁴⁷, avaliando se dados matriciais suportados pela GDAL, possam ser armazenados no banco de dados Spatialite e acessados também no Global Mapper e QGIS. O arquivo escolhido para o armazenamento no Spatialite (RasterLite) será a ortofoto da Emplasa, 3313-1.tif.

A figura 56 ilustra o exposto:

Figura 56. Esquema de integração de dados da GDAL/OGR, Spatialite e SIG, em sistemas livres e proprietários.

⁴⁷ Rasterlite é uma biblioteca *open source* que provê recursos para a criação de pirâmides, alteração de resolução e conversão de formatos de dados matriciais no Spatialite, por meio de aplicativos específicos. A GDAL se comunica com a Rasterlite; esta, oferece o suporte *raster* do Spatialite. Mais informações em: <<http://www.gaia-gis.it/gaia-sins/rasterlite-docs/rasterlite-man.pdf>>. Acesso em 12 mar. 2015.

Ademais, é importante mencionar que o Spatialite é a base para o novo formato GeoPackage, estabelecido pelo OGC para compartilhamento de dados geográficos, sobretudo entre dispositivos móveis. Isto significa que avanços na integração de dados entre a OGR e o Spatialite podem, teoricamente, ser revertidos ao novo formato.

Como estabelecem os desenvolvedores:

“GeoPackage is the modern alternative to formats like SDTS and Shapefile. At its core, GeoPackage is simply a SQLite database schema. If you know SQLite, you are close to knowing GeoPackage. Install Spatialite – the premiere spatial extention to SQLite – and you get all the performance of a spatial database along with the convenience of a file-based data set that can be emailed, shared on a USB drive or burned to a DVD.

GeoPackage was carefully designed this way to facilitate widespread adoption and use of a single simple file format by both commercial and open-source software applications — on enterprise production platforms as well as mobile hand-held devices. GeoPackage is a standard from the Open Geospatial Consortium [...].” (Fonte:)

A adoção do GeoPackage está sendo gradual, embora sistemas como o ArcGIS e a própria GDAL/OGR, já suportem algumas especificações estabelecidas pelo OGC para o novo formato.

5.1 Virtual OGR e Spatialite: acesso a dezenas de formatos vetoriais

Os grandes sistemas proprietários como ArcGIS, MapInfo e Global Mapper, por exemplo, oferecem suporte cada vez maior (em leitura e escrita), a diferentes formatos de dados geoespaciais - muitos deles, também reconhecidos (plena ou parcialmente) pela GDAL/OGR.

A possibilidade de armazenar dados no Spatialite por meio da biblioteca, pode estender àqueles e demais SIG, a capacidade de acesso a um conjunto maior de dados geoespaciais.

Em nosso exemplo, iremos inserir dados do MDC da Prefeitura de São Paulo, em CAD, no Spatialite. O comando é:

```
ogr2ogr -f SQLite -dsco SPATIALITE=YES -explodecollections 3313-164.sqlite
3313-164.DXF -lco SPATIAL_INDEX=YES
```

O parâmetro “-lco SPATIAL_INDEX=YES”, faz com que a OGR crie o índice espacial (baseado em R-Tree) no arquivo gerado, o que aumenta a velocidade de consultas e recuperação gráfica das geometrias em SIG, principalmente quando grandes bases de dados são envolvidas.

O R-Tree é o índice espacial padrão da SQLite/Spatialite:

“An R-Tree is a special index that is designed for doing range queries. R-Trees are most commonly used in geospatial systems where each entry is a rectangle with minimum and maximum X and Y coordinates. Given a query rectangle, an R-Tree is able to quickly find all entries that are contained within the query rectangle or which overlap the query rectangle. This idea is easily extended to three dimensions for use in CAD systems.” (SQLite R-Tree:)

A figura 57 ilustra o MDC armazenado no Spatialite e carregado no Global Mapper:

Figura 57. Arquivo com dados DXF encapsulados no Spatialite e carregados no Global Mapper.

Os metadados do arquivo Spatialite indicados pelo Global Mapper, estão a seguir:

```

FILENAME=C:\Users\Nader\Desktop\Doc\3313-164.sqlite
DESCRIPTION=entities (C:\Users\Nader\Desktop\Doc\3313-164.sqlite)
AREA COUNT=16
AREA VERTEX COUNT=80
LINE COUNT=4754
LINE VERTEX COUNT=36986
POINT COUNT=467
MESH COUNT=0
UPPER LEFT X=323989.069
UPPER LEFT Y=7393761.609
LOWER RIGHT X=324775.569
LOWER RIGHT Y=7393097.039
WEST LONGITUDE=46° 43' 28.3413" W
NORTH LATITUDE=23° 33' 24.8387" S
EAST LONGITUDE=46° 43' 00.3285" W
SOUTH LATITUDE=23° 33' 46.7463" S
UL CORNER LONGITUDE=46° 43' 28.0593" W
UL CORNER LATITUDE=23° 33' 24.8387" S
UR CORNER LONGITUDE=46° 43' 00.3285" W
UR CORNER LATITUDE=23° 33' 25.1456" S
LR CORNER LONGITUDE=46° 43' 00.6092" W
LR CORNER LATITUDE=23° 33' 46.7463" S
LL CORNER LONGITUDE=46° 43' 28.3413" W
LL CORNER LATITUDE=23° 33' 46.4393" S
PROJ_DESC=UTM Zone -23 / SAD69_BR / meters
PROJ_DATUM=SOUTH AMERICAN 1969 (BRAZIL)
PROJ_UNITS=meters
COVERED AREA=0.523 sq km
CODE_PAGE=65001 (UTF-8)

```

Como se pode perceber, o arquivo carregado no Global Mapper perdeu elementos da legenda e apresentou problemas com relação à codificação (caracteres especiais).

O arquivo SpatialLite não conseguiu ser visualizado no QGIS, mesmo após a operação de reescrita do arquivo com o recurso ***explodecollections*** da OGR.

Para analisarmos se era um problema da OGR ou do QGIS, utilizamos o aplicativo gráfico que acompanha o SpatialLite para o carregamento do DXF, conforme indica a figura 58:

Figura 58. Aplicativo SpatiaLite GUI, *open source*, com o módulo para o armazenamento do DXF no formato SpatiaLite.

Após o procedimento, o QGIS conseguiu reconhecer o SpatiaLite gerado, conforme indica a figura 59:

Figura 59. Armazenamento do DXF no SpatiaLite. Notar que o SpatiaLite GUI tentou preservar os *layers* originais do arquivo CAD.

Notar que após os procedimentos de reescrita e explosão dos blocos, os elementos da legenda se tornaram visíveis no QGIS.

Os problemas com os caracteres especiais dos textos e a perda de elementos da legenda, apresentados pelo Spatialite carregado no Global Mapper, levaram-nos a indagar se o problema na conversão estava na OGR, dado que em testes anteriores obtivemos resultados semelhantes, com os formatos de intercâmbio.

Contudo, a abertura quase completa do arquivo no Global Mapper mostrou que a OGR conseguiu fazer a conversão, ao passo que o QGIS não conseguiu recuperar graficamente o mapa no formato de banco de dados.

O sucesso da utilização do Spatialite GUI na conversão e posterior carregamento no QGIS, mostrou a deficiência deste SIG na recuperação de dados CAD armazenados no formato Spatialite.

5.2 Armazenamento de dados matriciais no Spatialite pelas bibliotecas GDAL e Rasterlite

A GDAL, a partir da versão 1.7, permite armazenar, converter formatos, alterar a resolução e aplicar pirâmides em dados matriciais no banco de dados Spatialite. A seguir, exemplificaremos a utilização da GDAL na gravação de dados no SGBD Espacial:

Inicialmente, é preciso criar o banco de dados Spatialite que será utilizado para receber as ortofotos, mediante o seguinte comando da GDAL⁴⁸ com a respectiva resposta do sistema:

```
gdal_translate -of Rasterlite 3313-1.tif  
RASTERLITE:ortofotos.sqlite,table=raster -co DRIVER=JPEG  
Input file size is 5500, 4167  
0...10...20...30...40...50...60...70...80...90...100 - done.
```

Apesar de originalmente estar em formato TIFF, forçamos a conversão do arquivo 3313-1 para JPEG⁴⁹ quando do carregamento no Spatialite, a fim de diminuir o espaço usado em disco por meio de uma compressão na imagem original (de 64,8 MB originais, o arquivo no formato de banco de dados passou para 8 MB).

Quando carregado no QGIS, o arquivo apresentou, entre outros, os seguintes metadados:

48 No momento de escrita desta proposta, a versão da GDAL/OGR utilizada é a 1.11.1 de 24/09/2014 e a do QGIS é a 2.6 (Brighton).

49 Por meio do parâmetro '-co DRIVER=JPEG', em que a qualidade da imagem é mantida, por padrão, em 75% da original.


```

Driver
GDAL provider
Rasterlite
Rasterlite
Descrição do registro
C:/Users/Nader/Desktop/Rasterlite/ortofotos.sqlite
TILE_FORMAT=JPEG
Sistema de Referência Espacial da Camada
+proj=utm +zone=23 +south +ellps=GRS67 +towgs84=-57,1,-41,0,0,0,0 +units=m
+no_defs
Extensão da camada (fonte de projeção original da camada)
321321.1200000001117587,7393032.2000000020489097 :
324621.1200000001117587,7395532.4000000022351742

```

É interessante notar que a imagem conservou seu DATUM e coordenadas UTM originais, mesmo com a conversão do formato TIFF para JPEG. O QGIS, por sua vez, reconhece o arquivo Rasterlite por meio da GDAL, como indica a figura 61:

Figura 61. Ortofoto da Emplasa, apresentada no SIG na escala 1:5000, armazenada e processada no Rasterlite.

Notar que, apesar da compressão imposta pelo formato JPEG, a imagem ainda continua com boa resolução.

O Global Mapper não conseguiu interpretar os dados matriciais armazenados no SpatialLite; reconheceu, apenas, a extensão em coordenadas (retângulos envolventes) da ortofoto, como grides vetoriais. A figura 62 ilustra o exposto:

Figura 62. Arquivo SpatialLite com dados matriciais armazenados, visualizado no Global Mapper.

Este SIG reconheceu apenas a extensão de coordenadas, recuperadas graficamente como polígonos.

No exemplo anterior, foi inserida apenas uma ortofoto no SGBD Espacial, tendo em vista a restrição do recurso **gdal_translate** para o gerenciamento de muitos arquivos.

Para ultrapassar essa limitação, existe o comando **gdalbuildvrt** em que é possível, inicialmente, mosaicar as imagens individuais, criando um arquivo único (no formato Virtual da GDAL) para depois processá-lo e inserí-lo no Spatialite. Posteriormente, basta repetir o comando **gdal_translate** em conjunto com o Rasterlite para o arquivo virtual criado. Os processos estão ilustrados a seguir:

Passo 1:

```
gdalbuildvrt mosaico_raster.vrt *.tif
```

Passo 2:

```
gdal_translate -of Rasterlite mosaico_raster.vrt  
RASTERLITE:ortosmosaicadas.sqlite,table=source -co DRIVER=JPEG
```

Adicionalmente, é possível utilizar somente a biblioteca Rasterlite para processar e armazenar arquivos matriciais no banco de dados Spatialite. Os passos são indicados a seguir:

Passo 1: criar um banco de dados Spatialite vazio. Este processo precisa ser auxiliado pelo aplicativo “spatialite” (compilado para o sistema operacional) e digitado no console (ou graficamente pelo Spatialite GUI):

```
spatialite ortofotos.sqlite
```

Passo 2: os arquivos matriciais que serão inseridos no Spatialite deverão estar na mesma pasta dos aplicativos Rasterlite e acionados pelo seguinte comando:

```
rasterlite_load -d ortofotos.sqlite -T raster -D . -v50
```

Passo 3: Construção de pirâmides, isto é, as representações em baixa resolução para otimizar a performance de carregamento da imagem no SIG:

```
rasterlite_pyramid -d ortofotos.sqlite -T raster -v
```

Passo 4: construção do mosaico *raster*, ou seja, o conjunto das imagens matriciais a serem inseridas no SGBD Espacial:

```
rasterlite_topmost -d ortofotos.sqlite -T raster -v
```

Passo 5: carregar no QGIS o Spatialite criado com as ortofotos armazenadas, conforme indica a figura 63:

⁵⁰ O parâmetro '-T raster' indica o nome que será atribuído à base matricial no banco de dados; o “-D” indica que os dados matriciais no formato GeoTIFF serão processados na pasta, e; o “-v”, que a biblioteca pode processar e alimentar o banco de dados com as imagens.

Figura 63. Ortofotos mosaicadas no QGIS.

A RasterLite processou as imagens, unindo e convertendo-as para o formato JPEG, com resolução de 75% em relação às originais. Isto permitiu que o banco de dados ocupasse um espaço em disco 70% menor, em comparação aos arquivos TIFF originais.

Em nossos testes, o QGIS começa a apresentar dificuldades no gerenciamento de dados, a partir de quinze ortofotos em GeoTIFF carregadas no sistema simultaneamente (isto varia de computador a computador, porque envolve aspectos como processador, quantidade de memória RAM acessada pelo SIG, existência de placa de vídeo dedicada, etc). Contudo, depois de inseridas no SpatialLite pela RasterLite, o QGIS conseguiu facilmente gerenciar mais de mil ortofotos.

5.2.1 Conclusão do armazenamento do formato GeoTIFF no RasterLite

A implementação do SpatialLite no Global Mapper, reconhece apenas dados vetoriais, apresentando problemas no carregamento de dados matriciais

armazenados no formato. Isto indica que a RasterLite ainda não é suportada no Spatialite implementado neste SIG.

O QGIS, devido ao apoio da GDAL, consegue reconhecer dados vetoriais e matriciais (RasterLite) armazenados no SGBD Espacial.

CONCLUSÃO

A utilização da GDAL/OGR na linha de comando superou, em eficiência nas conversões e em número de recursos, sua implementação gráfica nas diferentes versões do QGIS analisadas, indicando que a metodologia apresentada nesta pesquisa pode, efetivamente, beneficiar usuários e instituições que utilizam este e demais SIG. Uma vez que os aplicativos GDAL/OGR invocados pela linha de comando são instalados juntamente ao QGIS, sua utilização se torna ainda mais simplificada aos profissionais que lidam com a Cartografia, a Geografia e o Geoprocessamento.

De forma geral, todos os formatos vetoriais analisados apresentaram boa recuperação em SIG, após o emprego da OGR e a conversão dos dados originais na estrutura CAD. Os principais problemas nas conversões estavam relacionados ao elemento *bloco* - muito utilizado no universo CAD -, mas gerenciado de maneira parcial pela biblioteca. O recurso *explodecollections* da OGR apresentou performance inferior à explosão de blocos, comparado ao sistema CAD utilizado nos testes - o que não prejudicou os principais parâmetros (geométricos, topológicos, de atributos e referência espacial) analisados nesta pesquisa. Basicamente, apenas os elementos condensados no bloco CAD (relacionados às legendas), apresentaram pequenos problemas com textos (recuperação gráfica e codificação dos caracteres especiais) em SIG. Os demais conteúdos dos mapas (geométricos, topológicos, atributos e referência espacial) foram carregados corretamente, após as conversões. A prévia explosão de blocos no formato DXF, em associação posterior com os aplicativos de conversão da OGR garantiu, por sua vez, uma melhor integração de dados CAD migrados para ambiente SIG, em comparação a ação da biblioteca isoladamente.

No modelo matricial, mesmo com suporte parcial pela GDAL, o formato ArcInfo Grid foi escrito de maneira correta no GeoTIFF. Isto permitiria, por exemplo, que o sistema proprietário Global Mapper acessasse o MNT Ikonos utilizado nos testes, uma vez que este SIG não suporta o formato ESRI ArcInfo Grid - apenas se o sistema ArcGIS estiver instalado na mesma máquina do aplicativo.

Como vimos teoricamente, o armazenamento de dados em formatos de intercâmbio no SGBD Espacial poderia solucionar problemas de integração de dados em SIG. Na prática, entretanto, isto só acontece se a conversão de dados (pré-existent) para o SGBD Espacial for bem-sucedida, o que depende de uma correta implementação dos recursos de gravação nos aplicativos empregados na conversão. O sucesso na escrita no SGBD espacial depende, portanto, do mesmo processo utilizado entre os formatos de intercâmbio de dados: a conversão sintática.

A escrita do DXF para o Spatialite pela OGR foi, no geral, satisfatória, apresentando pequenos problemas com a recuperação de elementos da legenda condensados no bloco do formato DXF (o que era esperado, uma vez que a OGR possui apenas suporte parcial a blocos CAD).

Pela GDAL, a gravação do GeoTIFF no Spatialite foi satisfatória; contudo, a possibilidade de recuperação gráfica destes dados em SIG, depende do suporte à RasterLite, ainda ausente em alguns sistemas, entre eles, o Global Mapper aqui testado.

Verificamos, finalmente, que a hipótese inicial - a GDAL/OGR seria suficiente para os processos de conversão sintática de dados geoespaciais -, foi confirmada e concluímos que a biblioteca contribui efetivamente para a integração de dados em SIG.

REFERÊNCIAS BIBLIOGRÁFICAS

- BORGES, K. A. V. **Modelagem de dados geográficos - uma extensão do modelo OMT para aplicações geográficas**. Belo Horizonte: Fundação João
- PINHEIRO, 1997. Dissertação de mestrado, Escola de Governo, 1997. Disponível em: <>. Acesso em: 12 de março de 2015.
- BURROUGH, P. A. **Principles of Geographical Information Systems for Land Resources Assessment**. Oxford University Press, 1989. 194 p.
- CÂMARA, G.; CASANOVA, M.; DAVIS, C.; QUEIROZ, G.R.; VINHAS, L. **Banco de Dados Geográficos**. Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/bdados/>>. Acesso em: 12 de março de 2015.
- CÂMARA, G.; DAVIS, C.; MONTEIRO, A.M.V. **Introdução à Ciência da Geoinformação**. Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/introd/>>. Acesso em: 12 de março de 2015.
- DECKER, D. **GIS DATA SOURCE**. USA. 2001. 193p.
- CARNEIRO, E. L. N. C.; CÂMARA, G.; NAMIKAWA, M. L. **Estudo Sobre Armazenamento de Modelagem Digital de Terreno em Banco de Dados Geográficos**. INPE, 2008. São Paulo.
- CASANOVA, M. A.et. al. (2005) “Integração e interoperabilidade entre fontes de dadosgeográficos”, Bancos de Dados Geográficos. Rio de Janeiro, p. 305- 340, EditoraMundoGeo.
- DAVIS, C. **Geometria Computacional para SIG**. Geoprocessamento: Teoria e Aplicações. INPE, 2001.
- DAVIS, C. GIS Interoperável e Distribuído: Arquitetura e Aplicações. **Anais do IVWorkshop em Tratamento de Imagens**, NPDI/DCC/ICEx/UFMG, p. 26-37, Junho de 2003.
- De SMITH, M. J; GOODCHILD, M.F; LONGLEY, P. **Geospatial Analysis – a comprehensive guide**. Second Edition. Disponível em: <>. Acesso em: 12 de março de 2015.
- DOLVE, O.; POMPEO, J.N. **Fundamentos de Matemática Elementar - Geometria Espacial, Posição e Métrica**. Editora atual, 1985.
- FERREIRA, K.F.; QUEIROZ, G.R. **Tutorial sobre Banco de Dados Geográficos**. GeoBrasil 2006. Disponível em: <http://www.dpi.inpe.br/TutorialBdGeo_GeoBrasil2006.pdf>. Acesso em: 12 de março de 2015.

FERREIRA, K.F.; QUEIROZ, G.R.; PAIVA, J.A.; SOUZA, R.C.M.; CÂMARA, G. **Arquitetura de Software para a construção de Banco de Dados Geográficos com SGBD Objeto-Relacionais**. INPE, São Paulo. Disponível em: <http://www.dpi.inpe.br/gilberto/papers/paper_sbdb2002.pdf>. Acesso em: 12 de março de 2015.

FILHO, B.S.S. **Modelagem de Dados Espaciais**. Curso de Especialização em Geoprocessamento. Departamento de Cartografia. Centro de Sensoriamento Remoto. UFMG. 2001.

FILHO, J.L.; IOCHPE, C. **Um Estudo sobre Modelos Conceituais de Dados para Projeto de Bancos de Dados Geográficos**. s/data.

Geospatial World. Edição: February 2015, Issue 07, ISSN 2277-3134, Página 14.

GOODCHILD, M.; EGENHOFER, M.J.; FEGEAS, R. Interoperating GISs: Panel on computational implementations of geographic concepts. In: **INTERNATIONAL CONFERENCE & WORKSHOP ON INTEROPERATING GEOGRAPHIC**

INFORMATION SYSTEMS. Santa Bárbara, 1997. Anais eletrônicos. Disponível em: <>. Acesso em: 12 de março de 2015.

GUESSER, F. **Figuras Geométricas**. Instituto Federal de Santa Catarina, 2011. Disponível em: <>. Acesso em 12 de março de 2015.

HARVEY, F.; KUHN, W.; PUNDT, H.; BISHR, Y.; RIEDEMANN, C. Interoperability: A central issue for sharing geographic information. **The Annals of Regional Science**. Springer-Verlag, 1999.

HORA, A.C. **Mapeamento de esquemas conceituais geográficos para esquemas gml e esquemas físicos de bancos de dados espaciais**. Dissertação de mestrado. UFMG, 2010.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Noções básicas de Cartografia**. Número 8. Rio de Janeiro, 1999. 130p.

JÚNIOR, P.D.L. **GeoBR: intercâmbio sintático e semântico de dados espaciais**. Dissertação de Mestrado. INPE, 2002.

JUNIOR, C.A.D; SOUZA, L.A.D; BORGES, K.A.V. **Disseminação de dados geográficos na Internet**. INPE, São Paulo. Disponível em: <>. Acesso em: 12 de março de 2015.

LIBAULT, A. **Geocartografia**. 1975.

LIMA, P. **GeoBR Intercâmbio de dados espaciais: modelos, formatos e conversores**. São José dos Campos, 1v. Dissertação (Mestrado) - INPE, 2002. Disponível em: < Acesso em: 25 de setembro de 2013.

LONGLEY, P. A.; GOODCHILD, M.; MAGUIRE, D.J.;RHIND, D. W. **Geographic Information Systems and Science**, 2005.

MOHAMMADI, H; , RAJABIFARD, A; WILLIAMSON, I. Development of an interoperable tool to facilitate spatial data integration in the context of SDI. **International Journal of Geographical Information Science**. 16 de março de 2010.

NADER, R.S. **O servidor de mapas da Prefeitura de Taboão da Serra - SP: uma proposta de implantação**. Dissertação de mestrado. 2008, USP. 114p.

PEREZ MACHADO, R.P. **Integração geo-escalar: proposta de uma metodologia de identificação de mudanças no uso, ocupação e cobertura do solo aplicando a tecnologia geoespacial**. Tese de Livre Docência. 2012, USP. 178p.

SILVA, A B. **Sistemas de Informações Georreferenciadas. Conceitos e Fundamentos**. Editora Unicamp. Campinas, 2003. 236p.

SILVA, R. **Banco de Dados Geográficos: uma análise das arquiteturas dual (SPRING) e integrada (Oracle Spatial)**. Dissertação de Mestrado. São Paulo. USP, 2002. 137 p.

STEINIGER, S.; BOCHER, E. An overview on current free and open source desktop GIS developments, **International Journal of Geographical Information Science** (2009).

STILLWELL, J.; GEERTMAN, S.; OPENSHAW, S. **Developments in Geographical Information and Planning**. New York, 1999.

THEOBALD, D. M. **Topology revisited: representing spatial relations**. Natural Resource Ecology Laboratory, Colorado State University. USA, 6 April 2001.

TIMBÓ, M. A. **Cartomensura**. Departamento de Geografia. UFMG. 2001.

VACCARI, L.; SHVAIKO, P.; MARCHESE, M. An emergent semantics approach to semantic integration of geo-services asnd geo-metadata in Spatial Data Infrastructures. **International Journal of Spatial Data Infrastructures**, v.3, 2008.

WORBOYS, M.; DUCKHAM, M. **GIS: a computing perspective**. CRC PRESS. 2004. 426p.

Documentação Técnica:

Descrição Técnica de bibliotecas no sistema operacional Linux. Disponível em: <<http://tldp.org/HOWTO/Program-Library-HOWTO/index.html>>. Acesso em 12 mar. 2015.

Descrição Técnica do formato ASCII. Disponível em: <<http://www.asciitable.com>>. Acesso em 12 mar. 2015.

Descrição Técnica do formato CityGML. Disponível em: <<http://www.opengeospatial.org/standards/citygml>>. Acesso em 12 mar. 2015.

Descrição Técnica do formato ESRI Shapefile. Disponível em: <<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>>. Acesso em 12 mar. 2015.

Descrição Técnica do formato ESRI Multipatch. Disponível em: <<http://www.esri.com/library/whitepapers/pdfs/multipatch-geometry-type.pdf>>. Acesso em 12 mar. 2015.

Descrição técnica do formato GeoPackage. Disponível em: <<http://www.geopackage.org/>>. Acesso em 19 mar. 2015.

Descrição Técnica do formato GeoTIFF:

- Disponível em: <<http://trac.osgeo.org/geotiff/>>. Acesso em 12 de mar. 2015.
- Disponível em: <<http://www.awaresystems.be/imaging/tiff/faq.html>>. Acesso em 12 mar. 2015.
- Disponível em: <<http://www.opengeospatial.org/projects/groups/geotiffswg>>. Acesso em 12 mar. 2015.

Descrição Técnica do formato KML: Disponível em: <<http://www.opengeospatial.org/standards/kml>>. Acesso em 12 de mar. 2015.

Descrição Técnica do MapInfo Universal Translator: Disponível em: <http://reference.mapinfo.com/software/mapinfo_pro/english/9.0/UT_UG.pdf>. Acesso em 12 mar. 2015.

Descrição Técnica do formato ZIP. Disponível em: <<https://www.pkware.com/support/zip-app-note/>>. Acesso em 12 mar 2015.

EPSG (*European Petroleum Survey Group*). Disponível em: <<http://spatialreference.org/>>. Acesso em 24 jun. 2015.

ESRI Shapefile Technical Description. An ESRI White pape. Disponível em: <<http://dl.maptools.org/dl/shapelib/shapefile.pdf>>. Acesso em 12 mar. 2015.

FME BLOG. Disponível em: <<http://blog.safe.com>>. Acesso em 12 mar. 2015.

FME Readers and Writers 2012. Disponível em: <<http://docs.safe.com/fme/pdf/FMEReadersWriters.pdf>>. Acesso em 01 out. 2013.

GDAL/OGR. Disponível em: <<http://www.gdal.org>>. Acesso em 9 mar. 2015.

GML Foundation Project. Developing and Managing GML Application Schemas. British Columbia, 15/05/2003. Disponível em: <http://www.ogcnetwork.net/system/files/GML_Best_Practices_Guide.pdf>. Acesso em 27 ago. 2014.

GRASS WIKI. Disponível em: <<http://grasswiki.osgeo.org/>>. Acesso em 24 jun. 2015.

INDE. Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais (ET-EDGV), versão 2.1.3, 2008.

OGC Coordinate Transformation Service. Disponível em <<http://www.opengeospatial.org/standards/ct>>. Acesso em 24 jun. 2015.

OpenGIS Implementation Standard versão for Geographic Information - Simple Feature Access - Part 1, versão 1.2.1. OGC, 28/05/2011.

Open Source Geospatial Foundation (OSGEO). Disponível em <<https://trac.osgeo.org/proj/>>. Acesso em 24 jun. 2015.

Plano de Ação da INDE. Disponível em: <<http://www.concar.gov.br/arquivo/PlanoDeAcaoINDE.pdf>>. Acesso em 12 mar. 2015.

QGIS 2.0. A gentle introduction to GIS. Disponível em: <http://docs.qgis.org/2.0/en/docs/gentle_gis_introduction/>. Acesso em 12 mar. 2015.

Rasterlite:

1) Disponível em: <<http://www.gaia-gis.it/gaia-sins/rasterlite-docs/rasterlite-man.pdf>>. Acesso em 12 mar. 2015.

2) Disponível em: <http://www.gdal.org/frmt_rasterlite.html>. Acesso em 12 mar. 2015.

Universidade Federal Fluminense (UFF). Sistema Geodésico Brasileiro (SGB). Cadastro territorial multifinalitário. 06/11/2012.

Spatialite. Disponível em: <<http://www.gaia-gis.it/gaia-sins/spatialite-manual-2.3.1.html>>. Acesso em 12 mar. 2015.

SQLite R-Tree: Disponível em: <<https://www.sqlite.org/rtree.html>>. Acesso em 21 mar. 2015.

Understanding Topology and Shapefiles. ArcUser, 2011. Disponível em: <<http://www.esri.com/news/arcuser/0401/topo.html>>. Acesso em 12 mar. 2015.

ICoachMath: Leading to Excellence in Math. Disponível em:
<<http://www.icoachmath.com/>>. Acesso em 12 mar. 2015.

Artigos

Geospatial World Annual Readers' Survey 2014. Disponível em
<<http://geospatialworld.net/>>. Acesso em 12 mar. 2015.

GEOeduc. **Pesquisa aponta o QGIS como o software livre de geoprocessamento mais usado.** Disponível em
<<http://www.geoeduc.com/pesquisa-aponta-o-qgis-como-o-software-livre-de-geoprocessamento-mais-usado/>>. Acesso em 05 mai. 2015.

IBGE. **Mudança do referencial geodésico.** Disponível em:
<<http://www.ibge.gov.br/home/geociencias/geodesia/pmrg/faq.shtm>>. Acesso em 24 jun. 2015.

Decretos

BRASIL. Decreto Federal nº 6.666 de 27 de Novembro de 2008. Disponível em:
<http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Decreto/D6666.htm>. Acesso em 12 mar. 2015.

Curso On-line

Geospatial Intelligence & the Geospatial Revolution. PennState University. Janeiro e fevereiro de 2015. Disponível em: <<https://pt.coursera.org/course/geoint>>. Acesso em 12 mar. 2015.