

Estruturação e validação de dados geográficos em ambiente orientado a objeto do Sistema *Gothic*

Joel Borges dos Passos
Rafael Barboza de Carvalho
Alex de Lima Teodoro da Penha
Leandro Luiz Silva França

Diretoria de Serviço Geográfico do Exército – DSG
QG Ex – Bloco “F” – 2º Pav – SMU– Brasília-DF– CEP: 70.630-901
{joel.passos, rafael.barboza, dapenha.alex, franca.leandro}@eb.mil.br

Abstract. The geometric validation of spatial data on object oriented environment is the process that determines whether the values from the database are complete and logically consistent. In the production line of the Geographical Service Of Brazilian Army (Diretoria de Serviço Geográfico-DSG) this process (phase) is accomplished through the Gothic environment, which consists of a system for processing, handling and storage of geospatial data. The purpose of this paper is to present the procedure adopted in the execution of this phase, using as object of study the features of ten cartographic sheets, scale 1:25.000, belonging to the Cartographic Mapping Project of Bahia (Brazilian state). In the course of this work are presented in the introduction, a brief history and basic concepts related to this phase, as well as, in the methodology, the description for preparation Gothic's environment and the sequence of operations. This sequence presents routines which verifies the topological rules internally built in the Gothic, describing the conformity with the domain values of the data accordingly to the foreseen for the modelling Vector Geospatial Data Structuring -EDGV. In the analysis and results are presented the mean of non-conforming objects, identified in the validation process, as well as the percentage after correction. Finally, at the end of the work it is verified that the phase of validation through object-oriented environment ensures the logical consistency while maintaining the quality of the data produced and made available by DSG.

Palavras-chave: validation, cartographic production, database, object technology, Gothic, validação, produção cartográfica, banco de dados, tecnologia de objetos.

1. Introdução

A demanda por informações geoespaciais atualizadas é cada vez mais crescente, pois a sociedade em geral utiliza essas informações como subsídio à tomada de decisão, baseada em uma nova arquitetura tecnológica, econômica, social, ambiental, política, organizacional e de gestão coletiva em um processo de reestruturação global (CONCAR, 2008). Para que se atenda à necessidade de produtos cartográficos de qualidade é preciso executar processos computacionais que garantam a consistência lógica (manutenção de relações lógicas, como arranjo conceitual, domínio e topologia) dos dados. Uma das fases, do processo de produção cartográfica, que proporciona essa consistência é chamada de estruturação e validação.

Segundo Saldanha (2005), a validação de banco de dados geográficos é o processo que determina se os valores do banco de dados estão completos e logicamente consistentes. Ela consiste de várias etapas, incluindo checagens lógicas e análise de inconsistências. Neste contexto a consistência lógica é determinada pelas características topológicas dos dados, conformidade com os valores de domínio dos dados e consistência das classes de objetos com o modelo conceitual.

Para executar essa validação os sistemas baseados em tecnologia de ambiente Orientado a Objeto são adequados para o aperfeiçoamento dos processos de produção cartográfica. Conforme explica Saldanha (2005), num modelo de dados orientado a objetos um banco de dados é considerado como uma coleção de objetos¹ do mundo real, ou seja, permitem que as

¹ Objeto é uma instância de uma classe; modela uma entidade do mundo real e é implementada como uma entidade computacional que encapsula estado e operações e responde aos pedidos por serviços (SILVA, 2002).

feições sejam representadas por objetos com comportamentos modelados da forma como são conhecidos na realidade.

O modelo de dados atual para a cartografia brasileira está definido na Especificação Técnica para Estruturação de Dados Geoespaciais Vetoriais (ET-EDGV). Esta especificação define uma modelagem de dados geoespaciais adequada para o uso em Sistemas de Informações Geográficas (SIG) (CONCAR, 2008).

No Brasil a Diretoria de Serviço Geográfico (DSG) do Exército Brasileiro tornou-se a instituição pioneira no uso de estrutura de dados geoespaciais vetoriais modelada com técnica de orientação a objetos. A implementação da modelagem em ambiente orientado a objetos foi possível somente após a adoção do sistema *Gothic*² (SALDANHA, 2005).

Na linha de produção desta instituição, é utilizado o Banco de Dados Geográfico Orientado a Objetos do sistema *Gothic* para realização de algumas fases, entre elas, a fase de validação. De acordo com Saldanha (2005), o sistema *Gothic* é um conjunto de ferramentas para construir e processar aplicações para informações relacionadas com o espaço. Estas aplicações capacitam a captura, armazenamento, pesquisa e a manipulação de dados espaciais.

A DSG, atualmente, desenvolve vários trabalhos para o Sistema Cartográfico Nacional, entre eles destaca-se o mapeamento da base cartográfica na escala de 1:25.000 e 1:50.000 do estado da Bahia (Projeto de Mapeamento da Bahia). Neste projeto a DSG, por meio do 3º Centro de Geoinformação - 3º CGEO, antiga 3ª Divisão de Levantamento - 3ª DL, utiliza o sistema *Gothic* para executar o processo de validação, geração da área contínua e a edição das cartas topográficas.

Diante do exposto, este trabalho tem como objetivo principal apresentar os procedimentos de validação de dados geográficos em ambiente orientado a objeto do sistema *Gothic* e analisar seus resultados. Logo, são apresentadas ao longo deste trabalho as sequências de operações, com base em rotinas desenvolvidas para automatizar o processo, que devem ser executadas na fase de estruturação e validação no âmbito da DSG.

2. Área de Estudo

A área de estudo refere-se a dez folhas do Mapeamento Sistemático situadas no oeste do Estado da Bahia. A identificação das folhas, na escala de 1:25.000, de acordo com Índice de Nomenclatura e o Mapa Índice (MI), é apresentada na Tabela 1. Esta região teve o voo fotogramétrico executado no ano de 2010, aquisição vetorial de dados geoespaciais em 2014 e reambulação em 2015.

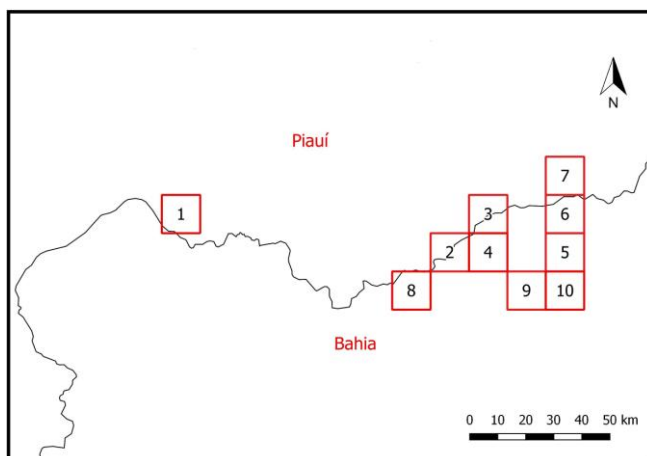


Figura 1 – Localização da área de estudo.

Tabela 1. Identificação das folhas

	ÍNDICE DE NOMENCLATURA	MAPA ÍNDICE
1	SC-23-X-D-I-3-NE	1510-3-NE
2	SC-23-X-D-III-3-SO	1512-3-SO
3	SC-23-X-D-III-3-NE	1512-3-NE
4	SC-23-X-D-III-3-SE	1512-3-SE
5	SC-23-X-D-III-4-SE	1512-4-SE
6	SC-23-X-D-III-4-NE	1512-4-NE
7	SC-23-X-D-III-2-SE	1512-2-SE
8	SC-23-X-D-V-2-NE	1586-2-NE
9	SC-23-X-D-VI-2-NO	1587-2-NO
10	SC-23-X-D-VI-2-NE	1587-2-NE

² Consiste de um sistema de tratamento, manipulação e armazenamento de dados geoespaciais, conhecido como LAMPS2 e um Banco de Dados Orientado a Objeto (BDOO) (SALDANHA, 2005).

3. Metodologia de Trabalho

3.1 Preparo do ambiente *Gothic*

De acordo com a metodologia de validação, o preparo do ambiente *Gothic* compreende a reunião dos recursos necessários à realização dos processos de estruturação e validação. Estes recursos constituem-se das seguintes atividades:

- a) Reunião dos insumos disponíveis que abrangem a região de trabalho.
- b) Criação do projeto. A criação de um projeto na plataforma *Gothic* consiste na definição de alguns parâmetros básicos, como sistema de projeção, datum, zona ou fuso UTM, hemisfério e escala da folha, de tal forma que cada *dataset*³ possa ser criado dentro de um projeto específico.
- c) Criação dos *datasets* individuais. Esta etapa constitui a criação dos *datasets* de cada folha do Mapeamento Sistemático. Esses *datasets* individuais correspondem a cada carta topográfica na respectiva escala.
- d) Instalação de Processos de Modelagem. A instalação do *schema*⁴ compreende a operação de criação de classes, métodos, atributos, relações, comportamentos e definições de armazenamento de cada classe da modelagem prevista na ET-EDGV.

Saldanha (2005) afirma que a forma de definição de um *schema* interativo é fornecida para permitir operações fáceis de definição de esquema, tais como adicionar novas classes ou atributos de objetos, ou definição de herança de classe.

As rotinas (regras) de definição de *schema* são implementadas na linguagem de programação LULL (*Lamps2 User Language*), exclusiva do ambiente *Gothic*, e definidas entre as classes segundo o estabelecido na modelagem da ET-EDGV. Estas rotinas são implementadas por meio de programação, no ato da instalação das classes e estabelecem os relacionamentos e quais as regras topológicas serão aplicadas entre elas.

Com os relacionamentos definidos o ambiente *Gothic* gera *links*⁵ para autoconexão de suas geometrias, as quais dependem de uma distância mínima entre as mesmas. A Tabela 2 apresenta essas distâncias, as quais são definidas de acordo com a escala de impressão (DSG, 2012). No caso deste trabalho, a escala de impressão das folhas é 1:25.000, logo a distância para aplicação da topologia é de 2 metros.

Tabela 2. Distâncias para aplicação da topologia em função da escala

Escala (1:)	Distância (m)
25:000	2
50.000	4
100.000	8
250.000	20

Fonte: DSG, 2012.

Concluindo a instalação, o *schema* da modelagem EDGV fica carregado em um *dataset*, estando em condições de receber os dados vetoriais oriundos do processo de aquisição.

e) Conversão e importação dos dados vetoriais. Para os dados serem validados no ambiente *Gothic*, necessitam passar por um processo de conversão de formatos, pois, o *Gothic* tem extensão própria (LULL). Para isso, utiliza-se o *software* FME (*Feature Manager Engine* – Máquina de Gerenciamento de Feições).

³ *Dataset* é um repositório de armazenamento do conjunto de informações geoespaciais no banco de dados *Gothic* (LASER-SCAN, 2000).

⁴ *Schema* é geralmente documentado em um dicionário de dados e fornece uma classificação lógica de objetos de banco de dados (ESRI, 2002).

⁵ *Link* é uma conexão semântica entre objetos. Em geral, um *link* é a instância de uma associação (SILVA, 2002).

Este *software* auxilia o *Gothic* nas operações de extração, transformação, importação e exportação de dados, possuindo ainda algumas funcionalidades para o tratamento preliminar dos mesmos quando isto se faz necessário.

Depois de realizar o preparo do ambiente *Gothic* e com os dados vetoriais carregados nos *datasets* segue-se à execução sobre os objetos do processo de estruturação e validação de acordo com as sequências de operações apresentadas a seguir.

3.2 Sequências de Operações

O processo de estruturação e validação consiste em criar nos *datasets* os *links* estabelecidos para o *schema* EDGV, os quais implementam as regras topológicas internamente construídas no *Gothic*, além de verificar a conformidade dos atributos das classes preenchidos nos objetos com as listas de valores de domínio previstas para a modelagem EDGV.

Com o objetivo de automatizar o processo de validação e estruturação, foram desenvolvidas, pela DSG, 16 rotinas que auxiliam a sequência de tarefas desenvolvidas pelo técnico validador. Estas rotinas são apresentadas a seguir.

- 1ª Rotina: Estruturar com topologia 20 vezes menor

Neste processo gera-se os elementos da topologia implementados pelo *Gothic* de acordo com o que foi definido para os relacionamentos da modelagem EDGV. Os elementos do *dataset* ficam com *links*, que têm como uma de suas finalidades atraírem outros elementos pertencentes à lista de classes definidas para se relacionar com a classe corrente, dentro de um raio de topologia, permitindo, assim, sua correta justaposição (Figura 2a e b) (DSG, 2012).

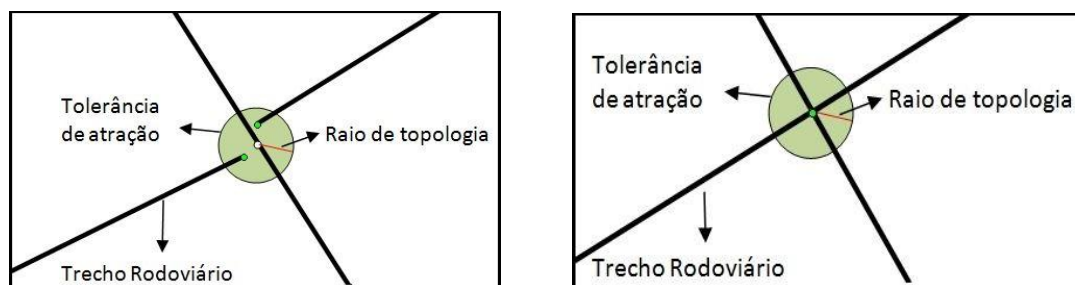


Figura 2 - (a) Elementos sem estruturação.

(b) Elementos após estruturação.

Para a aplicação dessas regras é definida uma distância mínima entre os objetos (ver Tabela 2). No entanto, para esta primeira rotina a distância adotada é 20 vezes menor que a prevista para a escala (DSG, 2012).

Após a execução desta rotina, caso haja objetos com erros de estruturação, serão gerados *markups*⁶ associados a estes objetos, que devem ser analisados e corrigidos. Neste trabalho entende-se por erro, em sentido mais amplo, como a não conformidade do elemento de acordo com as normas em vigor. Após as correções, deve-se executar o processo repetidamente até que não seja gerado mais nenhum *markup*.

- 2ª Rotina: Estruturar com topologia 20 vezes menor após correção

Após a execução da 1ª Rotina e das suas devidas correções é executada a 2ª Rotina com a finalidade de estruturar elementos que possivelmente não tenham sido estruturados na operação anterior ou que tenham sido alterados na correção.

- 3ª Rotina: Filtrar *links*

⁶ *Markups* contêm mensagens (*markup note*) com uma descrição sucinta do problema e, geralmente, um par de coordenadas que indicam a posição do erro (LASER-SCAN, 2000).

Esta rotina é executada com a finalidade de reduzir a quantidade de vértices dos *links*. O algoritmo usado neste processo de filtrar *links* é o de *Douglas Peucker*, originalmente desenvolvido por Douglas e Peucker (1973).

Vasconcelos e Sá (2012), explicam que este algoritmo está baseado na busca por pontos críticos (vértices), conforme Figura 3. Os pontos críticos que formam os *links* são aqueles que têm a maior distância perpendicular da linha de base para análise. A linha da primeira base será formada entre o primeiro vértice (âncora) e o último vértice (flutuante) da linha original. Em seguida, as distâncias perpendiculares são calculadas para todos os vértices entre estes dois vértices iniciais. Se qualquer uma dessas distâncias for menor que a tolerância atribuída, serão mantidos apenas os vértices de âncora e flutuante. Caso algum vértice exceda o valor da tolerância, este será mantido como um novo ponto crítico, ficando dessa forma, a linha original subdividida em duas seções e o processo se repete como se fossem duas feições distintas e assim sucessivamente até toda a extensão da feição.

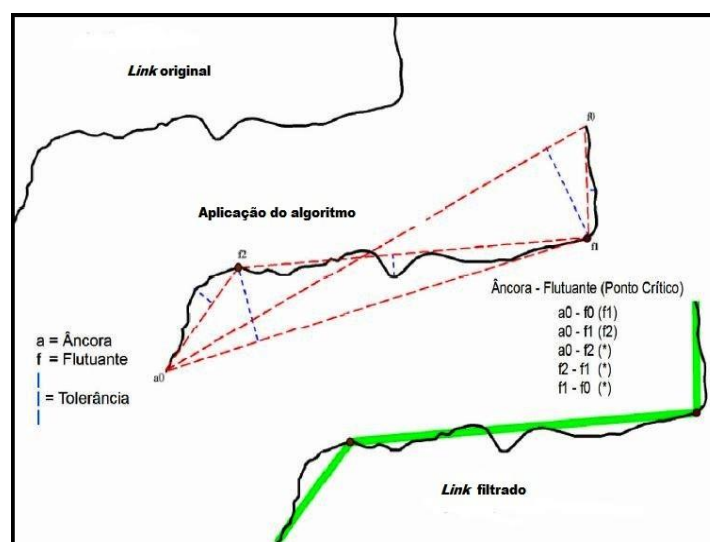


Figura 3 – Exemplo de funcionamento do algoritmo. Fonte: Adaptado de VASCONCELOS E SÁ, 2012.

Caso a remoção de vértices resulte em geometrias inválidas, *markups* serão associados aos *links* desses elementos. Deve-se, então, percorrê-los para efetuar as correções.

- 4ª Rotina: Estruturar com topologia 20 vezes menor após filtro

Esta rotina tem a finalidade de remover os *links* duplicados remanescentes das etapas anteriores. Deve-se percorrer os *markups* e corrigir os erros. Executar o processo novamente até que não seja gerado nenhum *markup*.

- 5ª Rotina: Checar loop

Este processo é utilizado para encontrar erros quando um elemento intercepta (faz o *loop*) ou sobrepõem ele mesmo em um determinado ponto. Este procedimento diferencia-se dos outros por que no momento em que um *markup* é gerado o *software* finaliza o processo, ou seja, é necessário que o operador corrija o erro e execute a rotina novamente até não ser gerado nenhum *markup*.

- 6ª Rotina: Filtrar *links* antes da topologia correta

Esta rotina é executada para depurar e prevenir possíveis erros de excesso de vértices que não foram acusados pelos processos anteriores.

- 7ª Rotina: Estruturar raio correto após filtro

A finalidade de rodar esta rotina é de renovar os *links* dos elementos já estruturados. Porém, para este processo, a estruturação tem topologia com raio de ação previsto para a escala de trabalho (2 metros, no caso da escala 1:25.000), permitindo uma maior conectividade entre os elementos.

- 8ª Rotina: Filtrar *links* topologia correta

Como no processo anterior os *links* são renovados com o raio da topologia correto para a escala de trabalho, então é necessário executar esta rotina para a eliminação de todos os vértices desnecessários ao *dataset* conforme explicado na 3ª rotina.

- 9ª Rotina: Checar *overlaps*

Nesta rotina, são verificados os elementos que possuem *links* compartilhados. Toda vez que dois elementos da mesma classe estiverem com um único *link*, ocorrendo uma sobreposição em uma determinada região, é gerado um *markup* neste local. Nesta rotina o *markup* é a sinalização de um possível erro, pois existem, em alguns casos, situações em que é permitida a sobreposição, conforme modelagem da EDGV.

- 10ª Rotina: Checar *loop*

Devido ao processo de filtrar *links* e estruturação com raio de topologia correto pode haver movimentação de vértices, logo, é necessário executar a rotina de checar *loop* novamente. Então, nesta fase repete-se a 5ª Rotina.

- 11ª Rotina: Seccionar linhas

Este processo gera interseções nos locais onde os objetos lineares se cruzam por meio da inserção de vértices quebrando-os. Ressalta-se que este processo não gera *markup*.

- 12ª Rotina: Estruturar raio correto

Nesta fase é repetido o processo da 7ª Rotina para renovar os *links* dos objetos que são alterados pela segmentação de linhas da 11ª Rotina.

- 13ª Rotina: Criar pontos de rede

Esta rotina é executada para interconectar, através dos pontos de rede, os elementos de dados (linhas) das seguintes classes: Trecho de Energia, Trecho Hidroviário, Trecho Duto, Trecho Drenagem, Trecho Rodoviário e Trecho Ferroviário.

- 14ª Rotina: Direcionar rios

Este processo direciona todos os trechos drenagem partindo da foz para a nascente. Geralmente ocorrem erros nesta fase quando algum ponto de rede, relacionado à hidrografia, são gerados com atributos errados ou quando existe ponto de ramificação que contornam as ilhas.

- 15ª Rotina: Checar *validate*

A execução desta rotina tem a finalidade de verificar se os valores dos atributos dos objetos do *dataset* estão preenchidos em conformidade com o previsto na modelagem EDGV.

- 16ª Rotina: Checar violação espacial

Esta é a última rotina da sequência de operação para estruturação e validação, nela é verificado se as regras topológicas de espacialidade dos objetos do *dataset* foram seguidas corretamente, conferindo as regras de *overlap*, *gap*, *intersect*, entre outras regras de validação.

Como exemplo, esta rotina verifica se duas vegetações estão sobrepostas ou se falta algum ponto de rede nas linhas.

Salienta-se que o operador, nesta sequência de operação, apenas prosseguirá para a próxima rotina quando todos os erros forem analisados e corrigidos, pois o erro de uma rotina poderá ocasionar vários erros nas próximas.

4. Análises e Resultados

Após a execução da sequência de operações, conforme apresentado na metodologia de trabalho, foi possível quantificar as “não conformidades” identificadas nas principais rotinas descritas (Tabela 3), de acordo com o *dataset* correspondente.

Tabela 3. Quantidade de *markups* encontrados nas principais rotinas em cada *dataset*.

Mapa Índice (MI) do Dataset	Total de objetos no Dataset	PRINCIPAIS ROTINAS						Proporção de <i>markups</i> em relação ao total de objetos (%)
		1 ^a	5 ^a	7 ^a	9 ^a	15 ^a	16 ^a	
1510-3-NE	6.818	38	16	04	11	21	58	2,1
1512-3-SO	3.413	23	07	01	12	08	22	2,1
1512-3-NE	3.962	18	23	02	8	13	16	2,0
1512-3-SE	2.861	08	15	03	03	06	09	1,5
1512-4-SE	1.251	06	02	00	00	03	05	1,2
1512-4-NE	1.284	09	00	00	03	05	10	2,1
1512-2-SE	3.130	13	03	00	06	10	21	1,7
1586-2-NE	3.640	09	04	00	02	07	15	1,0
1587-2-NO	1.711	10	05	01	03	06	18	2,5
1587-2-NE	4.218	17	11	02	12	19	15	1,8

Observa-se na Tabela 3 que existem *datasets* com uma quantidade de erros maior em todas as colunas em relação aos outros *datasets*. Essas diferenças variam de acordo com a quantidade de objetos existente em cada *dataset*, por exemplo, o MI 1510-3-NE possui 6.818 objetos, enquanto o MI 1512-4-SE possui apenas 1.251 objetos.

A quantidade de objetos que estes MIs possuem está relacionada diretamente com a região mapeada. O MI 1510-3-NE, por exemplo, engloba a área urbana da cidade de Caracol - PI, vários aglomerados rurais e a Serra das Confusões, ou seja, esta região tem muitas feições para serem representadas, enquanto o MI 1512-4-SE engloba uma região plana, com poucas curvas de nível e com apenas dois aglomerados rurais.

A última coluna da Tabela 3 mostra o percentual das não conformidades, encontradas na fase de validação, em relação ao total de objetos. Observa-se que os percentuais foram menores que 2,6%. Esse valor baixo é ocasionado pelo rigoroso controle de qualidade dos processos em fases anteriores à validação, na linha de produção da DSG.

Observa-se a ocorrência de alguns erros frequente em todos os *datasets* analisados neste trabalho. Por exemplo, para a 1^a Rotina (Estruturação) verificou-se que os principais erros ocorreram devido à aquisição de curvas de nível se auto-interceptando.

Já no processo de checar *loop* (5^a Rotina) a maior quantidade dos erros encontrados, em todos *datasets*, foram nos polígonos da categoria vegetação.

Enquanto na execução da 9^a Rotina (*overlaps*) verificou-se que a maior frequência de erros foi nos vetores da classe Trecho Drenagem que se auto-sobrepõem, ou seja, que passam por cima deles mesmos e nos vetores do Limite de Massas D'água que possuem dois vetores para uma única feição.

Para a 15^a Rotina (checar *validate*) verificou-se que os principais erros ocorreram por causa de valores errados nos pontos cotados, ou seja, o valor da cota não estava

correspondendo com as curvas de nível. Outro erro comum encontrado nesta rotina foi o não preenchimento de atributos obrigatórios de algumas classes.

E no processo de checar violação espacial (16ª Rotina), a qual verifica as regras topológicas de espacialidade dos objetos do *dataset*, foram detectados vários erros, principalmente de *intersect*, como exemplo, a curva de nível cruzar o limite geográfico ou sobrepor objetos da classe Trecho Massa D'água, ambos causando inconsistências topológicas associadas à geometria das feições.

Salienta-se que todas as não conformidades, encontradas no processo de validação, foram corrigidas, e as folhas saíram com 100% de conformidade.

5. Conclusões

Observou-se ao longo do desenvolvimento deste trabalho que as 16 rotinas de estruturação e validação aplicadas sobre aos objetos armazenados no BDOO garantem, após as correções, a consistência lógica dos dados. A validação identificou em média 1,8% de não conformidade no controle de qualidade interno e que, depois de corrigido, permitiu 100% de conformidade dos 32.288 objetos dos dez *datasets*.

Os processos garantem tanto a conferência dos atributos nas respectivas classes, para que nenhum atributo de classes deixe de ser preenchido com seu respectivo domínio, bem como a validação espacial, verificando erros de restrição espacial entre as classes, determinado pelos relacionamentos espaciais do modelo conceitual da ET-EDGV.

Portanto, ao validar a base cartográfica digital em um BDOO no sistema *Gothic*, tem-se maior segurança e integridade das relações entre os dados que a compõem, possibilitando eliminar multiplicidade de informações, além de garantir regras topológicas e proceder a correções cabíveis de feições inconsistentes, o que é fundamental para atingir o padrão de qualidade necessário para utilização dos dados em SIG, conforme estabelecido na ET-EDGV.

Referências Bibliográficas

- CÂMARA, G. **Análise de Arquiteturas para Bancos de Dados Geográficos Orientados a Objetos**. São José dos Campos, SP: USP, 1994. Tese de Doutorado.
- CONCAR - COMISSÃO NACIONAL DE CARTOGRAFIA. **Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais. ET-EDGV**. Brasília, 2008.
- DSG – DIRETORIA DE SERVIÇO GEOGRÁFICO. **Especificações Técnicas para a Aquisição de Dados Geoespaciais Vetoriais – ET-ADGV**, 1ª versão. Brasília, 2008.
- DSG – DIRETORIA DE SERVIÇO GEOGRÁFICO. **Metodologia de Validação Topológica do Conjunto de Dados Geoespaciais**, Brasília, 2012.
- DSG – DIRETORIA DE SERVIÇO GEOGRÁFICO **Especificação Técnica para Produtos de Conjuntos de Dados Geoespaciais (ET-PCDG)**, 1. Ed., 2. Rev., Brasília, 2015.
- ESRI – ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE. **ArcGIS User's Guide**. ESRI, 2002.
- LASER-SCAN. **Gothic Database Concepts v1.1**. Training Course. Laser-Scan, Cambridge, 2000.
- SALDANHA, MARCUS F. S. **Validação De Dados Geográficos Espaciais em Ambiente Orientado a Objeto**. Porto Alegre, RS: UFRGS, 2005. Dissertação de Mestrado.
- SILVA, A. **Dominando a Tecnologia de Objetos, UML-Java-C++**. Book Express, 2002.
- VASCONCELOS, T.L.; SÁ, L. A. C. M. **Generalização Cartográfica de Feições Lineares**. Anais IV Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação. Recife - PE, UFPE, 06- 09 de Maio de 2012.