

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  
**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**  
**Pós-graduação *Lato Sensu* em Desenvolvimento de Aplicações Web**

**Leonardo Cabral da Rocha Soares**

**ESTUDO DO PADRÃO GEOJSON: COMPARAÇÃO COM OUTROS FORMATOS E  
APLICAÇÃO EM UM PROJETO WEB.**

Belo Horizonte

2015

**Leonardo Cabral da Rocha Soares**

**ESTUDO DO PADRÃO GEOJSON: COMPARAÇÃO COM OUTROS FORMATOS E  
APLICAÇÃO EM UM PROJETO WEB.**

Trabalho de Conclusão de Curso de  
Especialização em Desenvolvimento de  
Aplicações Web como requisito parcial à  
obtenção do título de especialista.

Orientador(a): Marcos André Silveira Kutova

Belo Horizonte

2015

## RESUMO

A presente pesquisa tem como objetivo analisar o padrão **GeoJSON** para representação de geometrias espaciais. O formato será comparado com dois outros padrões também amplamente utilizados com o mesmo objetivo, o padrão **ShapeFile** e o padrão **OpenGIS KML Encoding Standard (KML)**. Serão apresentadas as principais características dos três formatos, as formas utilizadas por eles para representar as geometrias, aplicações que já os utilizam e uma análise comparativa entre eles sob a ótica do desenvolvimento de aplicações para web. Será apresentado um estudo de caso no qual investigaremos os benefícios do uso do padrão **GeoJSON** na produção de mapas online por meio da incorporação do padrão na biblioteca **GeoPUCMinas**.

**Palavras-chave:** GeoJSON. KML. ShapeFile. Mapas. Web. SIG.

## SUMÁRIO

<b>1. Introdução.....</b>	<b>5</b>
1.1. Contexto.....	5
1.2. Situação problema.....	6
1.3. Hipóteses.....	6
1.4. Objetivo.....	6
1.5. Justificativa.....	7
<b>2. Referencial teórico.....</b>	<b>7</b>
<b>3. Metodologia.....</b>	<b>8</b>
<b>4. Desenvolvimento.....</b>	<b>8</b>
4.1. Introdução.....	9
4.2. GeoJSON.....	10
4.3. OpenGIS KML Encoding Standard (KML).....	13
4.4. ShapeFile.....	14
4.5. Comparação entre os formatos sob a ótica do desenvolvimento web.....	15
4.6. Estudo de caso – Utilização do padrão GeoJSON com a biblioteca GeopUCMinas.....	17
<b>5. Análise dos resultados.....</b>	<b>20</b>
<b>6. Conclusão.....</b>	<b>20</b>
<b>REFERÊNCIAS.....</b>	<b>22</b>

# 1. Introdução

## 1.1. Contexto

O processo de construção de mapas tem sido diretamente afetado pelo desenvolvimento de tecnologias computacionais, tornando-se o computador o principal instrumento na tarefa de produzir mapas e a Internet o principal veículo divulgador.

Usualmente um mapa para web é acessado em um conjunto de ferramentas e serviços que possibilitam a interação entre o usuário e o mapa. Estes recursos associados são desenvolvidos por diversas empresas e programadores, em diversos idiomas e utilizando diversas tecnologias. Entretanto, é uma necessidade recorrente que mapas produzidos sob essas diversas condições troquem informações entre si.

Para permitir o intercâmbio de dados entre diferentes sistemas de produção de mapas, é necessário a adoção de um padrão na representação da informação. Entre os modelos propostos para uso padrão na representação de dados geográficos está o GeoJSON.

O GeoJSON é “um formato de codificação de estruturas geográficas de dados. Um objeto GeoJSON pode representar uma geometria, uma característica, ou uma coleção de características.”<sup>1</sup> (BUTLER et al., 2008, tradução nossa). Uma estrutura de dados GeoJSON completa, é sempre um objeto nos termos do formato JSON, que é um formato leve para intercâmbio de dados computacionais, baseado em um subconjunto da linguagem de programação JavaScript, sendo distribuído como um texto completamente independente da linguagem, pois suas convenções são suportadas por diversas linguagens de programação. Estas propriedades fazem com que JSON seja um formato ideal para troca de dados e explicam porque ele tem sido cada vez mais adotado como um padrão no intercâmbio de informações web.

Apesar da existência de padrões para a representação das geometrias, sua adoção no desenvolvimento de um sistema *web* é opcional. Em alguns casos, particularidades do sistema impedem a utilização de um padrão

---

<sup>1</sup>GeoJSON is a format for encoding a variety of geographic data structures. A GeoJSON object may represent a geometry, a feature, or a collection of features.

reconhecido e acabam por criar uma metodologia própria para representação dos dados. Como exemplo, podemos citar a biblioteca GeoPUCMinas desenvolvida pelo programa de pós-graduação em Tratamento da Informação Espacial da PUC Minas, que utiliza JSON puro para representar as geometrias. Esta biblioteca foi utilizada para a produção do Mapeamento da Educação Superior em Minas Gerais (MAPES), cujo objetivo foi tornar as informações públicas sobre a Educação Superior acessíveis aos gestores das Instituições de Educação Superior (IES) e poderia ser utilizada na produção de diversos outros projetos, mas a não utilização de um padrão dificulta a interação com outros sistemas.

## **1.2. Situação problema**

A adoção do padrão GeoJSON contribui para o desenvolvimento de sistemas de informações geográficas?

## **1.3. Hipóteses**

- O padrão GeoJSON atende às necessidades atuais de representação de informações espaciais na web.
- A adoção do padrão GeoJSON contribui para a produção de mapas online.

## **1.4. Objetivo**

Descrever como a adoção do padrão GeoJSON poderá contribuir para o desenvolvimento de aplicações de informações geográficas.

Os objetivos específicos do presente projeto de pesquisa são:

- Estudar o padrão GeoJSON.
- Comparar o padrão GeoJSON com dois outros padrões de representação de informações espaciais.
- Investigar os benefícios do uso do padrão GeoJSON na produção de mapas online, por meio da incorporação do padrão na biblioteca GeoPUCMinas.

## 1.5. Justificativa

Com a evolução tecnológica, as pessoas passam a incorporar cada vez mais mapas a suas atividades diárias. O acesso universal à informação por meio da internet forma cada vez mais usuários dispostos a interagir com sistemas e ferramentas de exibição de geometrias.

Com o aumento do uso, aumenta-se também o mercado para a interação entre as diversas soluções computacionais desenvolvidas. A cada nova aplicação, uma nova necessidade de intercâmbio de dados surge e fortalece-se a necessidade de estabelecermos um padrão para representação desses dados.

## 2. Referencial teórico

O presente estudo sobre o formato GeoJSON tem como principal referência sua documentação oficial disponível no site [geojson.org](http://geojson.org). O formato é descrito como “um formato de codificação de estruturas geográficas de dados<sup>2</sup>.” (BUTLER et al., 2008, tradução nossa). A documentação apresenta detalhadamente as geometrias suportadas pelo formato e exemplos de sua utilização.

Por tratar-se de um formato baseado no padrão JSON “Uma estrutura de dados GeoJSON completa é sempre um objeto (em termos JSON).<sup>3</sup>” (BUTLER et al., 2008, tradução nossa) – a documentação deste formato também servirá como base a este estudo.

Para compararmos o formato GeoJSON, o presente estudo apresentará dois outros padrões de exibição de geometrias, o KML e o ShapeFile.

O padrão KML - O OpenGIS KML Encoding Standard (OGC KML)<sup>4</sup> - foi enviado pelo Google para o Open Geospatial Consortium (OGC) para ser adotado como um padrão internacional de exibição de geometrias. É um formato XML focado na representação de dados geográficos, incluindo anotação de mapas e imagens. Sob esse olhar, a própria documentação considera-o um complemento a diversos padrões OGC existentes

Desta perspectiva, é complementar a maioria dos formatos existentes nos padrões da OGC, incluindo o GML (Geography Markup Language), WFS (Web Feature Service) e WMS (Web Map Service). Atualmente, KML 2.2 utiliza certos elementos de geometria derivados de GML 2.1.2. Esses elementos incluem ponto, cadeia de linha, anel linear, e polígono.<sup>5</sup> (Em:

---

<sup>2</sup>is a format for encoding a variety of geographic data structures.

<sup>3</sup>A complete GeoJSON data structure is always an object (in JSON terms).

<sup>4</sup>Disponível em <<http://www.opengeospatial.org/standards/kml/>>

<<http://www.opengeospatial.org/standards/kml>>. Acesso em 01 de junho 2015, tradução nossa)

O ShapeFile é um formato de dados vetorial para representação de dados geospaciais desenvolvido e regulamentado pela Environmental Systems Research Institute (ESRI) e amplamente utilizado por softwares de representação de informações geográficas (GIS).

Além dos formatos apresentados, o presente trabalho trará um estudo de caso utilizando a biblioteca GeoPUCMinas elaborada pelo programa de pós-graduação em Tratamento da Informação Espacial da PUC Minas e utilizada na produção do Mapeamento da educação superior em Minas Gerais utilizando métodos e técnicas de análise espacial e Web GIS – MAPES. Para utilização da biblioteca será consultado a documentação da mesma disponível no site [www.kutova.com/geopucminas/](http://www.kutova.com/geopucminas/)

### **3. Metodologia**

De forma a atender os itens aqui propostos como objetivos, será utilizada uma metodologia qualitativa baseada em pesquisas bibliográficas e o estudo de caso da biblioteca GeoPUCMinas.

Em uma primeira fase, será feito um estudo do padrão GeoJSON e das vantagens da sua utilização em sistemas de geometrias web. Após esta fase inicial, os demais padrões de exibição de geometrias serão analisados e comparados com o padrão GeoJSON.

Concluídas as análises, prosseguiremos com um estudo de caso da GeoPUCMinas, onde serão investigados os benefícios que o uso do padrão GeoJSON pode acarretar na produção de mapas online, por meio da incorporação do padrão na biblioteca GeoPUCMinas.

### **4. Desenvolvimento**

---

<sup>5</sup>From this perspective, KML is complementary to most of the key existing OGC standards including GML (Geography Markup Language), WFS (Web Feature Service) and WMS (Web Map Service). Currently, KML 2.2 utilizes certain geometry elements derived from GML 2.1.2. These elements include point, line string, linear ring, and polygon.



#### 4.1. Introdução

O crescente uso de tecnologias computacionais mudou a forma de produzirmos e consumirmos informações geográficas. Uma nova categoria de software foi criada e desenvolvida especialmente com esta finalidade, os sistemas de informações geográficas (SIG/GIS). Para Burrough e McDonnell, um SIG

é um poderoso conjunto de ferramentas para coleta, armazenamento, recuperação, transformação e visualização de dados espaciais do mundo real para um conjunto de propósitos específicos. (Burrough; McDonnell, 1998, p.11, tradução nossa).

A existência de diversos SIGs torna desejável a interação entre eles. Para isso, separa-se os dados que representam as geometrias dos softwares que a interpretam. Fitz, ao discorrer sobre os SIGs cita a importância da comunicação entre os diversos sistemas

Um bom sistema de informações geográficas deve proporcionar a conversão, suportar a importação e permitir a exportação de diferentes formatos de arquivos. A enorme quantidade de softwares disponíveis no mercado traduz uma imensa gama de tipos de arquivos com peculiaridades mais ou menos significativas e eles relacionadas. Formatos híbridos e/ou específicos de cada software deverão ser passíveis de conversões para serem exportados para outros. (Fitz, Paulo Roberto. 2008, p. 86)

Percebendo a importância da adoção de um padrão para o desenvolvimento dos arquivos utilizados pelos SIGs, fundou-se em setembro de 1994 o Open Geospatial Consortium (OGC). A organização se descreve como

[...] um consórcio da indústria internacional composto por 518 companhias, agências governamentais e universidades que participam de um de consenso para desenvolver padrões de interfaces públicos. Os padrões OGC criam soluções interoperáveis de apoio que "geo-permitem" os serviços baseados em localização Web, wireless e demais soluções em TI. Os padrões capacitam os desenvolvedores a criarem tecnologias de representação de informações espaciais complexas de forma acessível e útil a todos os tipos de aplicações.<sup>6</sup> (Em: <<http://www.opengeospatial.org/ogc>>. Acesso em 20 de maio 2015, tradução nossa)

Assim, o OGC define padrões aos quais produtos e serviços deveriam se adequar para facilitar a interação entre diversas fontes de dados. Mas não é a única entidade a criar formatos para a representação de geometrias. Formatos fora do consórcio da OGC também foram desenvolvidos e acabaram tornando-se padrões

<sup>6</sup>The Open Geospatial Consortium (OGC) is an international industry consortium of 518 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OGC® Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream IT. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

de fato, um padrão que, embora não tenha sido desenvolvido por uma instituição padronizadora, conseguiu estabelecer-se no mercado e, por seu uso, ser reconhecido como um padrão.

## 4.2. GeoJSON

GeoJSON é um formato para representação de estruturas de dados geográficos baseado no JavaScript Object Notation (JSON), um formato leve e independente de linguagem para intercâmbio de dados na forma de texto. Foi desenvolvido em 2008 por meio de discussões da comunidade desenvolvedora de SIGs. O formato é amplamente aceito, sendo utilizado por serviços como *Twitter*, *GitHub* e diversas ferramentas de utilização de geometrias, tais como o *D3* e o *ArcGIS Online*.

Uma estrutura de dados GeoJSON completa é sempre um objeto consistindo de uma coleção de pares nome/valor, chamados de membros. Este objeto representa uma geometria, uma característica ou uma coleção de características. As geometrias suportadas pelo formato são:

- Ponto (Point): As coordenadas representam uma única posição.<sup>7</sup>

```
{ "type": "Point", "coordinates": [100.0, 0.0] }
```

- Linha (LineString): As coordenadas neste tipo de geometria devem ser um vetor com duas ou mais posições.

```
{ "type": "LineString",
  "coordinates": [ [100.0, 0.0], [101.0, 1.0] ]
}
```

- Polígono (Polygon): As coordenadas neste tipo de geometria devem ser um vetor de Anel Linear (LinearRing), uma linha com 4 ou mais coordenadas onde a primeira e a última posição são equivalentes, gerando uma forma fechada.

```
{ "type": "Polygon",
  "coordinates": [
    [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0,
0.0] ] ]
}
```

<sup>7</sup> As coordenadas são sempre x e y (longitude e latitude). Os espaços em branco utilizados nos exemplos não são significativos e foram adicionados apenas para facilitar a visualização.

- Multi-ponto (MultiPoint): As coordenadas neste tipo de geometria devem ser um vetor de pontos.

```
{ "type": "MultiPoint",
  "coordinates": [ [100.0, 0.0], [101.0, 1.0] ]
}
```

- Multi-linha (MultiLineString): As coordenadas neste tipo de geometria devem ser um vetor de linha.

```
{ "type": "MultiLineString",
  "coordinates": [
    [[100.0, 0.0],[101.0, 1.0]],
    [[102.0, 2.0],[103.0, 3.0]]
  ]
}
```

- Multi-polígono (MultiPolygon): As coordenadas neste tipo de geometria devem ser um vetor de polígonos.

```
{ "type": "MultiPolygon",
  "coordinates": [
    [[[102.0, 2.0],[103.0, 2.0],[103.0, 3.0],[102.0, 3.0],[102.0, 2.0]]],
    [[[100.0, 0.0],[101.0, 0.0],[101.0, 1.0],[100.0, 1.0],[100.0, 0.0]],
     [[100.2, 0.2],[100.8, 0.2],[100.8, 0.8],[100.2, 0.8],[100.2, 0.2]]]
  ]
}
```

- Coleção de Geometrias (GeometryCollection): Este objeto representa um vetor de geometrias. Cada objeto neste vetor é uma geometria.

```
{ "type": "GeometryCollection",
  "geometries": [
    { "type": "Point",
      "coordinates": [100.0, 0.0]
    },
    { "type": "LineString",
      "coordinates": [ [101.0, 0.0], [102.0, 1.0] ]
    }
  ]
}
```

Além das geometrias citadas, um objeto GeoJSON pode ser uma Feature que contém um objeto geométrico e propriedades adicionais, ou uma coleção de Features (FeatureCollection) que representa um vetor de Features.

```

{ "type": "FeatureCollection",
  "features": [
    { "type": "Feature",
      "geometry": {"type": "Point", "coordinates": [102.0, 0.5]},
      "properties": {"prop0": "value0"}
    },
    { "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": 0.0
      }
    },
    { "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],
            [100.0, 1.0], [100.0, 0.0] ]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": {"this": "that"}
      }
    }
  ]
}

```

### 4.3. OpenGIS KML Encoding Standard (KML)

Originalmente desenvolvido pela Google, o formato foi submetido ao OGC para ser adaptado e convertido em um padrão formalizado pela instituição. A documentação descreve o formato como

[...] um XML usado para codificar e transportar representações geográficas de dados para exibição em um navegador da terra. Simplificando: KML codifica o que mostrar em um navegador da terra, e como mostrá-lo. KML utiliza uma estrutura de tags com elementos aninhados e atributos e baseia-se no padrão XML.<sup>8</sup> (The OGC KML Standard, p. 18, tradução nossa)

<sup>8</sup>is an XML grammar used to encode and transport representations of geographic data for display in an earth browser. Put simply: KML encodes what to show in an earth browser, and how to show it. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard.

O formato é utilizado para exibir dados geográficos em navegadores da terra como o Google Earth e Google Maps. Diversos profissionais e instituições tem utilizado o formato para marcar locais e eventos (históricos e atuais). Organizações como a National Geographic e a UNESCO utilizam o formato para exibir conjuntos de dados globais.

Qualquer marcador<sup>9</sup> adicionado ao Google Earth é um elemento do tipo Ponto (Point) em um arquivo KML. O código KML para um marcador simples pode ser exemplificado por<sup>10</sup>

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"> <Placemark>
  <name>Simple placemark</name>
  <description>Attached to the ground. Intelligently places itself at the
height of the underlying terrain.</description>
  <Point>
  <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
</Point>
</Placemark> </kml>
```

Como podemos ver no exemplo, a primeira linha de um documento KML é um cabeçalho XML, a especificação diz que nenhum caractere pode aparecer antes desta linha. A segunda linha é uma declaração de namespace KML, também obrigatória em todo documento KML. Depois é criado um Placemark contendo nome e descrição e um ponto que representa a geometria que será exibida no mapa.

O padrão KML trabalha com as seguintes geometrias: Pontos (Point), Linhas (LinearString ou LinearRing), Polígonos (Polygon), Geometrias múltiplas (MultiGeometry) e Modelo (Model).

Para poder informar ao cliente web como desenhar a geometria armazenada, o padrão suporta, além das geometrias propriamente, diversas tags utilizadas para a formatação e personalização dos mapas.

Estruturas de dados complexas podem resultar em mais de um arquivo KML e nesse caso, eles podem ser unificados em um arquivo comprimido no formato KMZ.

#### 4.4. ShapeFile

O ShapeFile é um formato de dados vetorial para representação de dados geospaciais desenvolvido e regulamentado pela Environmental Systems Research Institute (ESRI). O formato é amplamente utilizado por softwares desktop de

<sup>9</sup>Um recurso que marca um local na superfície da terra usando um alfinete.

<sup>10</sup>Retirado de [https://developers.google.com/kml/documentation/kml\\_tut?hl=pt-br](https://developers.google.com/kml/documentation/kml_tut?hl=pt-br)

representação de informações geográficas tais como ArcGIS for Desktop, ArcGIS Explorer Desktop, Quantum GIS, SuperGIS e gvSIG. Sua popularidade entre SIGs é tão grande que o formato é considerado um formato universal de representação de geometrias.

A descrição técnica do formato, disponibilizada pela ESRI, apresenta a constituição do mesmo

um ESRI shapefile consiste de um arquivo principal, um arquivo de índice e uma tabela dBASE. O arquivo principal é um arquivo para acesso direto contendo registros de tamanho variáveis e em cada registro uma descrição da forma com uma lista de seus vértices. No arquivo de índice, cada registro contém o deslocamento correspondente ao registro no arquivo principal a partir do seu início. A tabela dBASE contém as características dos registros, uma ocorrência por registro. O relacionamento entre as geometrias e as características é de um-para-um e é baseado no número do registro. As características armazenadas na tabela dBASE devem ser cadastradas na mesma ordem em que aparecem no arquivo principal<sup>11</sup>. (ESRI Shapefile Technical Description, p. 6, tradução nossa)

Os arquivos do formato devem ser nomeados utilizando-se a convenção 8.3 (nomes de arquivos curtos) e todos devem possuir o mesmo prefixo. O arquivo principal possui o sufixo .shp, o arquivo de índice .shx e a tabela dBASE .dbf.

O formato possui 13 formas de representação de geometrias: Point, PolyLine, Polygon, MultiPoint, PointZ, PolyLineZ, PolygonZ, MultiPointZ, PointM, PolyLineM, PolygonM, MultiPointM e MultiPatch. Cada geometria é descrita na documentação, com seus respectivos pontos e configurações.

#### **4.5. Comparação entre os formatos sob a ótica do desenvolvimento web**

Entre os formatos aqui apresentados, embora seja o mais utilizado, o ShapeFile é o que menos vantagem apresenta para o desenvolvedor web. A criação dos arquivos que compõem o modelo não é tão simples como nos demais padrões apresentados. Por não se tratar de um texto puro e de fácil leitura para humanos sua criação exige aplicações próprias. A tecnologia utilizada na estruturação do formato não é um padrão amplamente utilizado por desenvolvedores (como são os formatos XML e JSON) e isso consumiria tempo de desenvolvimento, pois obrigaria os

---

<sup>11</sup>An ESRI shapefile consists of a main file, an index file, and a dBASE table. The main file is a direct access, variable-record-length file in which each record describes a shape with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBASE table contains feature attributes with one record per feature. The one-to-one relationship between geometry and attributes is based on record number. Attribute records in the dBASE file must be in the same order as records in the main file.

desenvolvedores a entenderem como manipular o formato em que os dados estão armazenados, ou a utilizarem uma API de terceiros. A própria ESRI disponibiliza uma API REST para permitir aos desenvolvedores trabalharem com o conteúdo do ArcGIS online em JavaScript.

O KML utiliza a anotação XML para exibir as geometrias, a maioria dos desenvolvedores está acostumado com este formato que é inteligível a seres humanos, o que torna sua edição e criação consideravelmente simples. Com o formato é possível adicionar às geometrias características de formatação, não se restringindo apenas aos dados geométricos. Além de ser um padrão oficializado pela OGC, o padrão tem o respaldo de haver sido desenvolvido pela Google e amplamente utilizado nos produtos da mesma.

O padrão GeoJSON utiliza o formato JSON, entre os padrões aqui apresentado ele é o que mais facilmente pode ser entendido por seres humanos, o que torna extremamente fácil sua criação e manutenção. Além disso, é um formato extremamente leve e nativo do JavaScript (embora sua utilização não requeira JavaScript exclusivamente).

Quando falamos de aplicações web modernas, todo desenvolvedor deve preocupar-se com os dados que estão sendo trafegados. É cada vez maior o número de usuários que utilizam dispositivos móveis para acessar os aplicativos e a transferência de dados gera custo. O formato GeoJSON consegue descrever as mesmas geometrias dos outros padrões utilizando um número menor de bytes.

Para exemplificar, comparemos a exibição de um ponto nos formatos KML e GeoJSON.

```
<kml xmlns="http://earth.google.com/kml/2.0">
  <Placemark>
    <name>Bora-Bora Airport</name>
    <Point>
      <coordinates>-151.752044,-16.443118</coordinates>
    </Point>
  </Placemark>
</kml>
```

```
{
  "type": "Feature",
  "geometry": {
```

```

    "type": "Point",
    "coordinates": [-151.752044, -16.443118]
  },
  "properties": {
    "name": "Bora-Bora Airport"
  }
}

```

Claramente, ambos os formatos são legíveis a olhos humanos, mas o segundo (GeoJSON) é mais simples o que facilita o entendimento. Se considerarmos como parâmetro avaliativo a quantidade de caracteres necessários para a representação das geometrias, o KML necessitou de 169 caracteres enquanto o GeoJSON precisou de 125 caracteres (em ambos os arquivos foram desconsiderados os espaços não significativos e o carácter de nova linha), economizando de 44 caracteres ou 26,03%.

Como o formato ShapeFile é composto por vários arquivos o parâmetro anterior – contagem simples de caracteres – não se aplica. Para uma análise satisfatória, comparamos seu tamanho em bytes. Convertemos o ShapeFile disponibilizado pelo Comitê da Bacia Hidrográfica do Rio São Francisco<sup>12</sup> para o formato GeoJSON, utilizando a ferramenta online da Ogr - ogr2ogr web client<sup>13</sup> e para KML, utilizando a ferramenta online da Mapsdata<sup>14</sup>. O arquivo original (em ShapeFile) possuía 17.652 bytes. Após as conversões, o arquivo GeoJSON ficou com 39.007 bytes e o arquivo KML ficou com 66.176 bytes, sendo o ShapeFile o menor entre eles. Entretanto, o formato ShapeFile está compactado, ao compactarmos os arquivos convertidos, o arquivo GeoJSON fica com apenas 9.714 bytes (44,07% menor) e o arquivo KML com 10.547 bytes (40,25% menor), ou seja, sob condições iguais de compactação o formato GeoJSON é que menos ocupa espaço na representação de geometrias.

#### 4.6. Estudo de caso – Utilização do padrão GeoJSON com a biblioteca GeoPUCMinas

O Programa de Pós-graduação em Tratamento da Informação Espacial (PPG-TIE) da PUC Minas, desenvolveu o Mapeamento da Educação Superior em Minas Gerais (**MAPES**<sup>15</sup>), cujo objetivo foi tornar as informações públicas sobre a Educação

<sup>12</sup>Disponível em <<http://www.saofrancisco.cbh.gov.br/Shapes/SHAPE.sigeo.adutora.pb.RN.zip>>

<sup>13</sup>Disponível em <<http://ogre.adc4gis.com/>>

<sup>14</sup>Disponível em <<http://www.mapsdata.co.uk/online-file-converter/>>

<sup>15</sup>Disponível em: <<http://www.kutova.com/mapes/>>



Superior acessíveis aos gestores das Instituições de Educação Superior (IES). O **MAPES**, apresenta um conjunto de mapas gerados a partir de dados extraídos do XII Censo Demográfico, realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em 2010, do Censo da Educação Superior de 2010, realizado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), e do Atlas do Desenvolvimento Humano no Brasil 2013, resultado da parceria entre o Programa das Nações Unidas para o Desenvolvimento (PNUD), o Instituto de Pesquisa Econômica Aplicada (IPEA) e a Fundação João Pinheiro (FJP).

Para a criação dos mapas, o **MAPES** utiliza uma biblioteca escrita em JavaScript, a **GeoPUCMinas**<sup>16</sup>. Os dados que serão apresentados nos mapas são armazenados em um objeto JSON na seguinte forma:

```
var dados = {  
  '310010': 6704,  
  '310020': 22690,  
  '310030': 13311,  
  ...  
};
```

Este objeto JSON pode ser escrito na própria página ou incorporado por meio de um arquivo externo. Cada item do objeto contém o código do município (de acordo com o IBGE) e o valor da variável medida para este município.

As geometrias que formam os municípios também são armazenadas utilizando-se o formato JSON. O arquivo contendo as geometrias é descrito em comentários no próprio código fonte (KUTOVA, 2013), sendo formado por três objetos. Cada qual contendo:

- Uma lista de mesorregiões contendo uma sublista para cada UF. O rótulo de cada sublista é a sigla da UF. Essa sublista é um conjunto de vetores, em que cada vetor armazena os dados de uma mesorregião. O rótulo de cada vetor é o código de 2 dígitos da mesorregião. Os elementos do vetor são (nessa ordem) o nome da mesorregião e a geometria (a geometria é um vetor contendo vários subvetores. Cada subvetor é um par de coordenadas).
- Uma lista de microrregiões contendo uma sublista para cada UF. O rótulo de cada sublista é a sigla da UF. Essa sublista é um conjunto de vetores, em que cada vetor armazena os dados de uma microrregião. O rótulo de cada vetor é

---

<sup>16</sup>Disponível em: <<http://www.kutova.com/geopucminas/>>

o código de 2 dígitos da microrregião. Os elementos do vetor são (nessa ordem) o nome da microrregião, o código da mesorregião e a geometria (a geometria é um vetor contendo vários subvetores. Cada subvetor é um par de coordenadas).

- Uma lista de municípios contendo uma sublista para cada UF. O rótulo de cada sublista é a sigla da UF. Essa sublista é um conjunto de vetores, em que cada vetor armazena os dados de um município. O rótulo de cada vetor é o código do município com 6 dígitos. Os elementos do vetor são (nessa ordem) o nome do município, o código da microrregião, o código da mesorregião, a área, a latitude da sede, a longitude da sede e a geometria (a geometria é um vetor contendo vários subvetores. Cada subvetor é um par de coordenadas).

Embora seja funcional e eficiente, o formato adotado pela biblioteca **GeoPUCMinas** foi desenvolvido especificamente para ela, não seguindo nenhum padrão formalizado. A ausência de um padrão dificulta o desenvolvimento de novas geometrias e impacta negativamente na utilização da biblioteca em outros projetos. Entretanto, o formato utilizado pode facilmente ser adaptado para o padrão GeoJSON. Como exemplo, a geometria que representa a cidade de Belo Horizonte no formato atual é:

```
'310620': [ "Belo Horizonte",'30','07',330.93,-19.817,-43.956,[[-43.975807,-19.781309],[-43.965279,-19.791332],[-43.959141,-19.778114],[-43.945808,-19.776907],[-43.943222,-19.797722],[-43.898167,-19.802158],[-43.891918,-19.814148],[-43.876667,-19.813475],[-43.861862,-19.827026],[-43.867722,-19.843212],[-43.856918,-19.857513],[-43.876804,-19.867126],[-43.882164,-19.861456],[-43.896225,-19.863712],[-43.898945,-19.874647],[-43.909168,-19.875757],[-43.905582,-19.885677],[-43.890499,-19.885902],[-43.870693,-19.927944],[-43.913082,-19.962671],[-43.939419,-19.974564],[-43.974998,-20.008549],[-43.996887,-20.037294],[-44.001831,-20.057035],[-44.013306,-20.059357],[-44.017139,-20.053503],[-44.010361,-20.040539],[-44.035946,-20.019897],[-44.061943,-19.973921],[-44.045277,-19.973307],[-44.035114,-19.983261],[-44.012417,-19.967798],[-44.013584,-19.955751],[-44.00539,-19.954639],[-44.023586,-19.943296],[-44.022141,-19.923717],[-44.028889,-19.915064],[-44.026333,-19.898531],[-44.012802,-19.881893],[-44.018864,-19.860561],[-44.008751,-19.853802],[-44.019836,-19.839712],[-44.016861,-19.831297],[-44.00536,-19.82102],[-44.007584,-19.813133],[-43.990139,-19.785503],[-43.975807,-19.781309]] ]
```

Uma possível adaptação para o padrão GeoJSON seria:

```

{"type":"FeatureCollection","features":[

{"type":"Feature","id":"310620","properties":{"name":"Belo Horizonte"},"geometry":
{"type":"Polygon","coordinates":[[[-43.975807,-19.781309],[-43.965279,-19.791332],[-43.959141,-
19.778114],[-43.945808,-19.776907],[-43.943222,-19.797722],[-43.898167,-19.802158],[-43.891918,-
19.814148],[-43.876667,-19.813475],[-43.861862,-19.827026],[-43.867722,-19.843212],[-43.856918,-
19.857513],[-43.876804,-19.867126],[-43.882164,-19.861456],[-43.896225,-19.863712],[-43.898945,-
19.874647],[-43.909168,-19.875757],[-43.905582,-19.885677],[-43.890499,-19.885902],[-43.870693,-
19.927944],[-43.913082,-19.962671],[-43.939419,-19.974564],[-43.974998,-20.008549],[-43.996887,-
20.037294],[-44.001831,-20.057035],[-44.013306,-20.059357],[-44.017139,-20.053503],[-44.010361,-
20.040539],[-44.035946,-20.019897],[-44.061943,-19.973921],[-44.045277,-19.973307],[-44.035114,-
19.983261],[-44.012417,-19.967798],[-44.013584,-19.955751],[-44.00539,-19.954639],[-44.023586,-
19.943296],[-44.022141,-19.923717],[-44.028889,-19.915064],[-44.026333,-19.898531],[-44.012802,-
19.881893],[-44.018864,-19.860561],[-44.008751,-19.853802],[-44.019836,-19.839712],[-44.016861,-
19.831297],[-44.00536,-19.82102],[-44.007584,-19.813133],[-43.990139,-19.785503],[-43.975807,-
19.781309]]] ] } }

```

Ao analisarmos o exemplo, podemos perceber que as adaptações necessárias são pequenas e o impacto no código fonte da biblioteca será mínimo pois assim como o formato atualmente utilizado, objetos GeoJSON são nativos da linguagem JavaScript e facilmente manipulados pela mesma.

Ao adotar o padrão GeoJSON, a criação de novas geometrias tornar-se-á um trabalho menos árduo, será possível adaptar facilmente geometrias desenvolvidas para outros sistemas, mesmo se, esses utilizarem um padrão diferente do GeoJSON pois existem diversos conversores entre os padrões mais amplamente utilizados.

## 5. Análise dos resultados

Embora ainda não seja padronizado pela Open Geospatial Consortium (OGC), o formato GeoJSON mostrou-se o mais vantajoso sob a ótica do desenvolvimento web. Além de ser o formato mais compacto, o fato de ser um objeto nativo na linguagem JavaScript facilita imensamente sua utilização em um ambiente de produção focada na web. Sua ampla aceitação no mercado de desenvolvimento de geometrias atual e a fácil conversão de outros padrões para este formato são outro fator que impactam positivamente na escolha da adoção deste padrão.

O motivo de não ser padronizado pelo OGC não ficou claro durante esta pesquisa. A organização não padroniza nenhum formato puramente texto embora esses formatos sejam reconhecidamente mais leves. O fato da organização, aparentemente, priorizar o formato XML em detrimento de formatos nativos em JavaScript também pode ser alvo de pesquisas futuras.

## **6. Conclusão**

O processo de construção de mapas tem sido diretamente afetado pelo desenvolvimento de tecnologias computacionais, e essas tecnologias tem sido diretamente afetadas pela evolução da web. Como consequência, um número cada vez maior de sistemas para representação de geometrias é voltado para a web. O desenvolvimento web necessita de padronização, pois como não temos controle sobre os milhares de desenvolvedores, se não houver um formato padronizado, os sistemas serão incapazes de se comunicar e crescerão isoladamente, sem a possibilidade de interação entre eles.

Sendo assim, padrões e formatos são estabelecidos para nortear o desenvolvimento das aplicações. Existe um grande número de padrões formalmente descritos pela Open Geospatial Consortium (OGC) para representar geometrias espaciais. Entre esses formatos, destacam-se o ShapeFile e o KML. Além dos formatos padronizados pelo OGC, a ampla utilização de um formato o torna um “padrão de fato”, é esse o caso do formato GeoJSON, que tem sido amplamente utilizado por empresas e desenvolvedores ao redor do mundo. Esses três padrões, aqui apresentados, atendem perfeitamente as necessidades atuais para representação de informações espaciais na web. Entre eles, a adoção do padrão GeoJSON é a que mais efetivamente contribui para o desenvolvimento de aplicações web por ser o formato mais facilmente legível, que menos espaço ocupa e por ser nativamente um objeto da linguagem JavaScript que estará presente em quase cem por cento das aplicações web.



## REFERÊNCIAS

- ArcGIS API for JavaScript.** Disponível em <[https://developers.arcgis.com/javascript/jssamples/portal\\_addshapefile.html](https://developers.arcgis.com/javascript/jssamples/portal_addshapefile.html)>. Acesso em 30 jun. 2015.
- Ajuda do ArcGIS Online.** Disponível em <<http://doc.arcgis.com/pt-br/arcgis-online/reference/shapefiles.htm>>. Acesso em 25 jun. 2015.
- An ESRI White Paper.** Disponível em <<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>>. Acesso em 26 jun. 2015.
- BURROUGH; McDONNELL, R. **Principles of Geographical Information Systems.** New York: Oxford University Press, 1998.
- Butler, Howard. Et al. **The GeoJSON Format Specification.** 2008. Disponível em <<http://geojson.org/>>. Acesso em 06 jun. 2015.
- DE MENDONÇA, André Luiz Alencar. **Avaliação de interfaces para mapas funcionais na web.** Tese de Doutorado. Universidade Federal do Paraná, 2009.
- ECMA INTERNATIONAL. **ECMA-404 The JSON Data Interchange Format. 1st Edition.** October 2013. <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>
- Environmental Systems Research Institute. **ESRI Shapefile Technical Description, The JavaScript Object Notation (JSON) Data Interchange Format, Technical Report, Internet Engineering Task Force, 2014. RFC 7159**
- FITZ, Paulo Roberto, **Geoprocessamento sem Complicação,** São Paulo, Ed. Oficina de Textos, 2008.
- FITZ, Paulo R. **Cartografia Básica (nova edição),** Ed. Oficina de Textos, 2008.
- Jim Detwiler, John A. **Reading Shapefiles Using JavaScript Libraries.** Disponível em <<https://www.e-education.psu.edu/geog863/node/1959>>. Acesso em 30 jun. 2015.
- KUTOVA, Marcos A. S. **Mapeamento da educação superior em Minas Gerais utilizando métodos e técnicas de análise espacial e Web GIS.** 2013. 254f. Tese (Doutorado) – Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-graduação em Geografia – Tratamento da Informação Espacial, Belo Horizonte.
- Open Geospatial Consortium. **OGC History (abbreviated).** Disponível em <<http://www.opengeospatial.org/ogc/history>>. Acesso em 10 jun. 2015.
- Open Geospatial Consortium. **OGC KML.** 2009. Disponível em <<http://www.opengeospatial.org/standards/kml>>. Acesso em 14 jun. 2015.

**Tutorial do KML.** Disponível em <[https://developers.google.com/kml/documentation/kml\\_tut?hl=pt-br](https://developers.google.com/kml/documentation/kml_tut?hl=pt-br)>. Acesso em 24 jun. 2015.

WAINER, Jacques. **Métodos de pesquisa quantitativa e qualitativa para a ciência computação.** in Tomasz Kowaltowski and Karin Breitman. (Org.). Atualização em informática 2007. Sociedade Brasileira de Computação e Editora PUC-Rio, 2007. p. 221-262. Disponível em: <<http://www.ic.unicamp.br/~wainer/papers/metod07.pdf>>. Acesso em 2 jun. 2015.