

Manual de Utilização da Biblioteca VisPublica



**Autores: Melise Paula
Fernanda Ribeiro
Aldeíde Brasil
Tulio Moraes
Bárbara Pimenta
Guilherme Ferreira**

Versão: 2.1.4

Manual de Utilização da Biblioteca VisPublica 2.1.4

Desenvolvido por:

**Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de
Engenharia - COPPE/UFRJ**

Programa de Engenharia de Sistema e Computação - PESC

Centro de Apoio a Políticas de Governo – CapGov



Manual de Utilização da Biblioteca VisPublica 2.1.4

Sumário

1. INTRODUÇÃO	4
2. INFORMAÇÕES GERAIS.....	4
2.1. BIBLIOTECA VISPUBLICA	4
2.2. DISPONIBILIZAÇÃO DA BIBLIOTECA	4
2.3. REQUISITOS DA BIBLIOTECA	5
3. USANDO A BIBLIOTECA	5
3.1. ADICIONANDO A BIBLIOTECA EM UM PROJETO WEB	5
3.2. UTILIZANDO A BIBLIOTECA EM UMA PÁGINA JSP	7

1. Introdução

O objetivo deste documento é apresentar o manual de utilização da Biblioteca VisPublica 2.1.4. A Biblioteca VisPublica foi desenvolvida em parceria entre a Universidade Federal de Itajubá (UNIFEI) e a COPPE/UFRJ, sendo um produto requerido pelo Ministério do Planejamento, Orçamento e Gestão.

2. Informações Gerais

2.1. Biblioteca VisPublica

A Biblioteca VisPublica é uma ferramenta desenvolvida em Java para converter arquivos XML em scripts de entrada. Nesta versão, os arquivos são convertidos para bibliotecas javascript capazes de gerar visualizações de dados em páginas web.

A Tabela 1 apresenta as técnicas desta versão.

Tecnologia	Técnica
The JIT	Sunburst
	Treemap
	Hypertree
	BarChart
	RGraph
HighCharts	Bar
	Column
	Line
	Pie
	Pirâmide Etária
	Gráfico Temporal
Gráfico Combinado	
SVG	Mapa

Tabela 1 – Técnicas disponíveis na versão 2.1.4.

Mais informações sobre as técnicas e seu funcionamento podem ser encontradas no portal VisPublica Web (<http://vispublica.gov.br/vispublica/>).

2.2. Disponibilização da Biblioteca VisPublica

A Biblioteca VisPublica se encontra disponível para download no link <http://vispublica.gov.br/vispublica/publico/bibliotecaVisPublica/vispublica.api-2.1.4.rar>, o arquivo compactado contém:

- Biblioteca VisPublica (vispublica.api.jar)
- Bibliotecas requeridas para a Biblioteca VisPublica
 - dom4j-1.6.1.jar
 - javaee-api-6.0.jar
 - xpp3_min-1.1.4c.jar
 - xstream-1.4.3.jar
- Pasta com recursos para construção das visualizações (resources_vispublica)
- Pasta com a documentação da Biblioteca VisPublica

2.3. Requisitos da Biblioteca VisPublica

O uso da Biblioteca VisPublica só é possível em aplicações web desenvolvidas com a tecnologia Java Server Pages (JSP). Para exibição das visualizações é necessário ter acesso à internet. Um dos seguintes navegadores pode ser utilizado para exibir as visualizações:

- Chrome 21 ou superior;
- Firefox 11 ou superior;
- Internet Explorer 9 ou superior.

3. Usando a Biblioteca VisPublica

3.1. Adicionando a Biblioteca VisPublica em um projeto web

O primeiro passo é adicionar o arquivo vispublica.api.jar e todos os arquivos do diretório /lib no classpath do projeto. Esses arquivos são responsáveis pela leitura do XML e conversão para os scripts das visualizações. Na Figura 1, é demonstrado um projeto exemplo criado na IDE NetBeans com as bibliotecas adicionadas ao classpath.

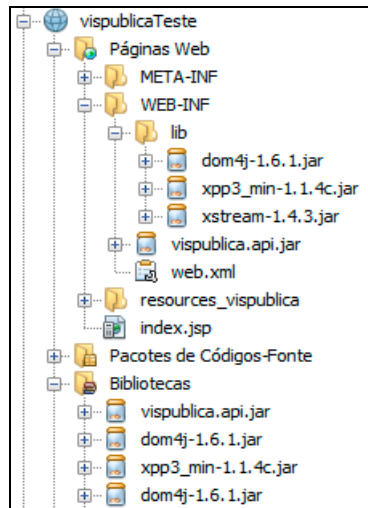


Figura 1 - Arquivos para Biblioteca VisPublica

Após incluir os jars ao classpath, é necessário adicionar o diretório /resources_vispublica no diretório /web do projeto. No projeto da Figura 1, o diretório /resources_vispublica pode ser visto no diretório “Páginas Web”.

Em seguida, deve-se atualizar o arquivo web.xml no diretório WEB-INF com o código mostrado na Figura 2. Caso esse arquivo não exista na aplicação, o mesmo deve ser criado.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0">
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <jsp-config>
    <!-- TAG DE CONFIGURAÇÃO DA API VISPUBLICA 2.0 -->
    <taglib>
      <taglib-uri>vispublicaTagLib.tld</taglib-uri>
      <taglib-location>/WEB-INF/vispublica.api.jar</taglib-location>
    </taglib>
  </jsp-config>
</web-app>
```

Figura 2 – Código a ser inserido no arquivo web.xml

Vale destacar que a tag <jsp-config> deve ser usada para a versão 3.0 do schema web.xml. Para versões anteriores não é necessário colocar a tag <jsp-config>.

3.2. Utilizando a Biblioteca VisPublica em uma página JSP

Para utilizar a Biblioteca VisPublica em uma página JSP deve-se inserir a diretiva `taglib` no início da página, conforme destaca a Figura 3.

```
<%@page import="br.gov.planejamento.vispublica.api.util.VisPublicaUtils"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!-- TAGLIB DA API VISPUBLICA 2.0 -->
<%@taglib uri="/vispublicaTagLib" prefix="vispublica" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Exemplo de Utilização da API VisPublica 2.0</title>
</head>
<body>
<% String x = VisPublicaUtils.getXMLFromURL(
    "http://localhost:8084/testeVisPublica/resources_vispublica/examples/treemap.xml"); %>
<vispublica:visualization id="visualizacao" visId="1001" xml="<%=x%>" />
</body>
</html>
```

Figura 3 – Configuração da diretiva `taglib`.

Além da `taglib`, no código html da página, é necessário incluir a tag `<vispublica:visualization>` para gerar a visualização (conforme destaque da Figura 3). Esta tag possui 3 atributos: `id`, `visId` e `xml`, sendo que todos são obrigatórios.

O atributo `id` é o identificador da visualização gerada na página html. Esse atributo pode receber qualquer valor textual, desde que não seja repetido em outra tag `<vispublica:visualization>`. A Figura 4 mostra um exemplo da tag `<vispublica:visualization>` com ids diferentes.

```
<%@page import="br.gov.planejamento.vispublica.api.util.VisPublicaUtils"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!-- TAGLIB DA API VISPUBLICA 2.0 -->
<%@taglib uri="/vispublicaTagLib" prefix="vispublica" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Exemplo de Utilização da API VisPublica 2.0</title>
</head>
<body>
<% String x = VisPublicaUtils.getXMLFromURL(
    "http://localhost:8084/testeVisPublica/resources_vispublica/examples/treemap.xml"); %>
<vispublica:visualization id="visualizacao" visId="1001" xml="<%=x%>" />
<vispublica:visualization id="visualizacao2" visId="1001" xml="<%=x%>" />
</body>
</html>
```

Figura 4 – Tag `<vispublica:visualization>` com ids diferentes.

O atributo `visId` é o código da técnica de visualização. A Tabela 2 apresenta os `ids` das técnicas disponíveis na Biblioteca VisPublica.

ID	Técnica de Visualização	Tecnologia
1001	Treemap	The JIT
1002	Sunburst	
1003	Hypertree	
1004	RGraph	
1005	BarChart	
3001	Bar	HighCharts
3002	Column	
3003	Line	
3004	Pie	
3005	Pirâmide Etária	
3006	Gráfico Temporal	
3007	Gráfico Combinado	
4001	Mapa	SVG

Tabela 2 – ID das técnicas de visualização disponíveis na Biblioteca VisPublica 2.1.4.

O atributo `xml` recebe o arquivo XML no formato de string, conforme mostra a Figura 5.


```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!-- TAGLIB DA API VISPUBLICA 20 -->
<%@taglib uri="/vispublicaTagLib" prefix="vispublica" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Exemplo de Utilização da API VisPublica 2.0</title>
  </head>
  <body>
    <!-- Exemplo do arquivo XML no formato de string -->
    <vispublica:visualization id="visualizacao" visId="1005"
      xml="<?xml version=\\"1.0\\" encoding=\\"UTF-8\\"?>
        <tecnica id=\\"1005\\">
          <metadados>
            <titulo>Número de Campi Avançados.</titulo>
            <grupos>
              <grupo tipo=\\"numerico\\">2009</grupo>
              <grupo tipo=\\"numerico\\">2010</grupo>
            </grupos>
          </metadados>
          <dados>
            <dado>
              <nome>Amazonas</nome>
              <valores>
                <valor grupo=\\"2009\\">300</valor>
                <valor grupo=\\"2010\\">750</valor>
              </valores>
            </dado>
            <dado>
              <nome>Ceará</nome>
              <valores>
                <valor grupo=\\"2009\\">260</valor>
                <valor grupo=\\"2010\\">1500</valor>
              </valores>
            </dado>
            <dado>
              <nome>Minas Gerais</nome>
              <valores>
                <valor grupo=\\"2009\\">130</valor>
                <valor grupo=\\"2010\\">600</valor>
              </valores>
            </dado>
            <dado>
              <nome>Rio Grande do Sul</nome>
              <valores>
                <valor grupo=\\"2009\\">477</valor>
                <valor grupo=\\"2010\\">884</valor>
              </valores>
            </dado>
            <dado>
              <nome>Goiás</nome>
              <valores>
                <valor grupo=\\"2009\\">100</valor>
                <valor grupo=\\"2010\\">500</valor>
              </valores>
            </dado>
          </dados>
        </tecnica>" />
  </body>
</html>

```

Figura 5 – Arquivo XML no formato de string.

A Figura 6 mostra a visualização do arquivo XML da Figura 5.

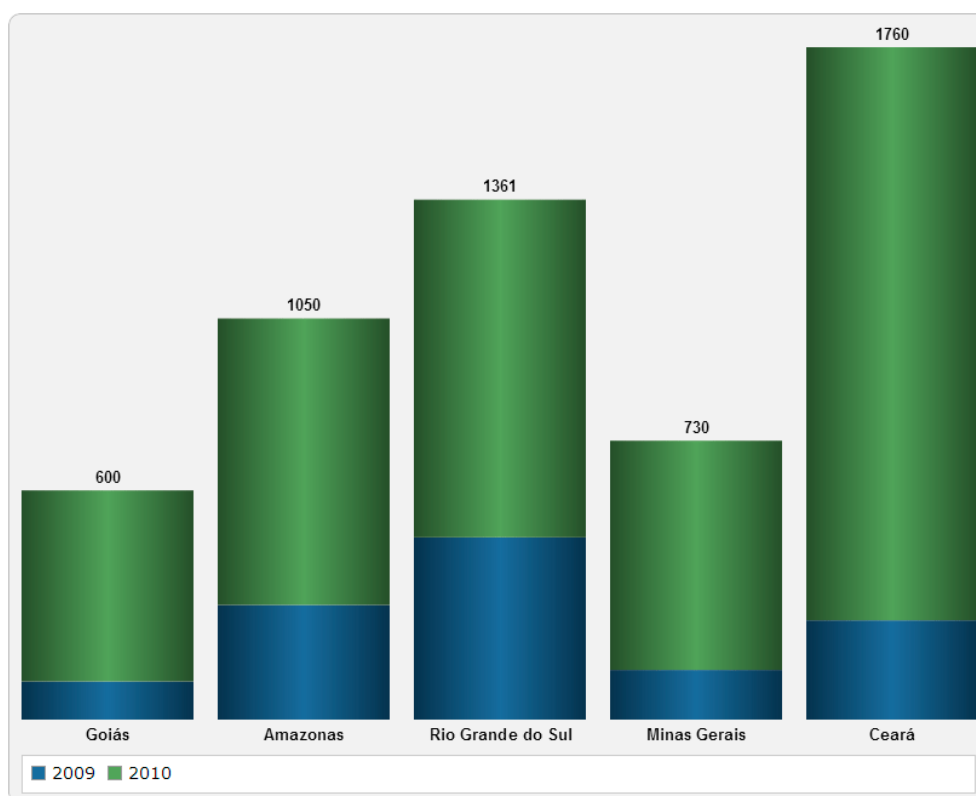


Figura 6 – Visualização gerada pelo XML da Figura 5.

Para auxiliar o usuário na conversão do arquivo XML em formato de string, a Biblioteca possui a classe `VisPublicaUtils.java` com os métodos `getFromFilePath(String path)` e `getFromURL(String url)` responsáveis por essa conversão.

O método `getFromFilePath(String path)` recebe como parâmetro o caminho do arquivo XML, como ilustrado na Figura 7.

```
<%@page import="br.gov.planejamento.vispublica.api.util.VisPublicaUtils"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!-- TAGLIB DA API VISPUBLICA 2. -->
<%@taglib uri="/vispublicaTagLib" prefix="vispublica" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Exemplo de Utilização da API VisPublica 2.0</title>
</head>
<body>
<!--
    CHAMADA DO MÉTODO QUE CONVERTE O ARQUIVO XML EM FORMATO STRING
    ESTE MÉTODO RECEBE COMO PARÂMETRO O CAMINHO DO ARQUIVO XML
-->
<% String xml = VisPublicaUtils.getXMLFromFilePath("C:/xml/treemapGoogle.xml"); %>
<vispublica:visualization id="visualizacao" visId="2001" xml="<%=xml%>" />
</body>
</html>
```

Figura 7 – Arquivo XML convertido em formato de string pelo método `getFromFilePath(String path)`.

A Figura 8 mostra a visualização gerada pelo arquivo XML da Figura 7.



Figura 8 – Visualização gerada pelo XML da Figura 7.

O método `getFromURL(String url)` recebe como parâmetro a URL do arquivo XML, como mostra a Figura 9. Note que o arquivo XML passado como parâmetro está dentro do diretório `/resources_vispublica/examples` (conforme destaca a Figura 9). O diretório `examples` contém exemplos de arquivos XML das técnicas disponíveis na Biblioteca VisPublica 2.1.4.

```
<%@page import="br.gov.planejamento.vispublica.api.util.VisPublicaUtils"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!-- TAGLIB DA API VISPUBLICA 2. -->
<%@taglib uri="/vispublicaTagLib" prefix="vispublica" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Exemplo de Utilização da API VisPublica 2.0</title>
</head>
<body>
<!--
CHAMA DO MÉTODO QUE CONVERTE O ARQUIVO XML EM FORMATO STRING
ESTE MÉTODO RECEBE COMO PARÂMETRO A URL DO ARQUIVO XML
-->
<% String xml = VisPublicaUtils.getXMLFromURL (
    "http://localhost:8080/vispublicaTeste/resources_vispublica/examples/hypertree.xml"); %>

<vispublica:visualization id="visualizacao" visId="1003" xml="<%=xml%>" />

</body>
</html>
```

Figura 9 – Arquivo XML convertido em formato de string pelo método `getFromURL(String url)`.

A Figura 10 mostra a visualização gerada pelo arquivo XML da Figura 9.

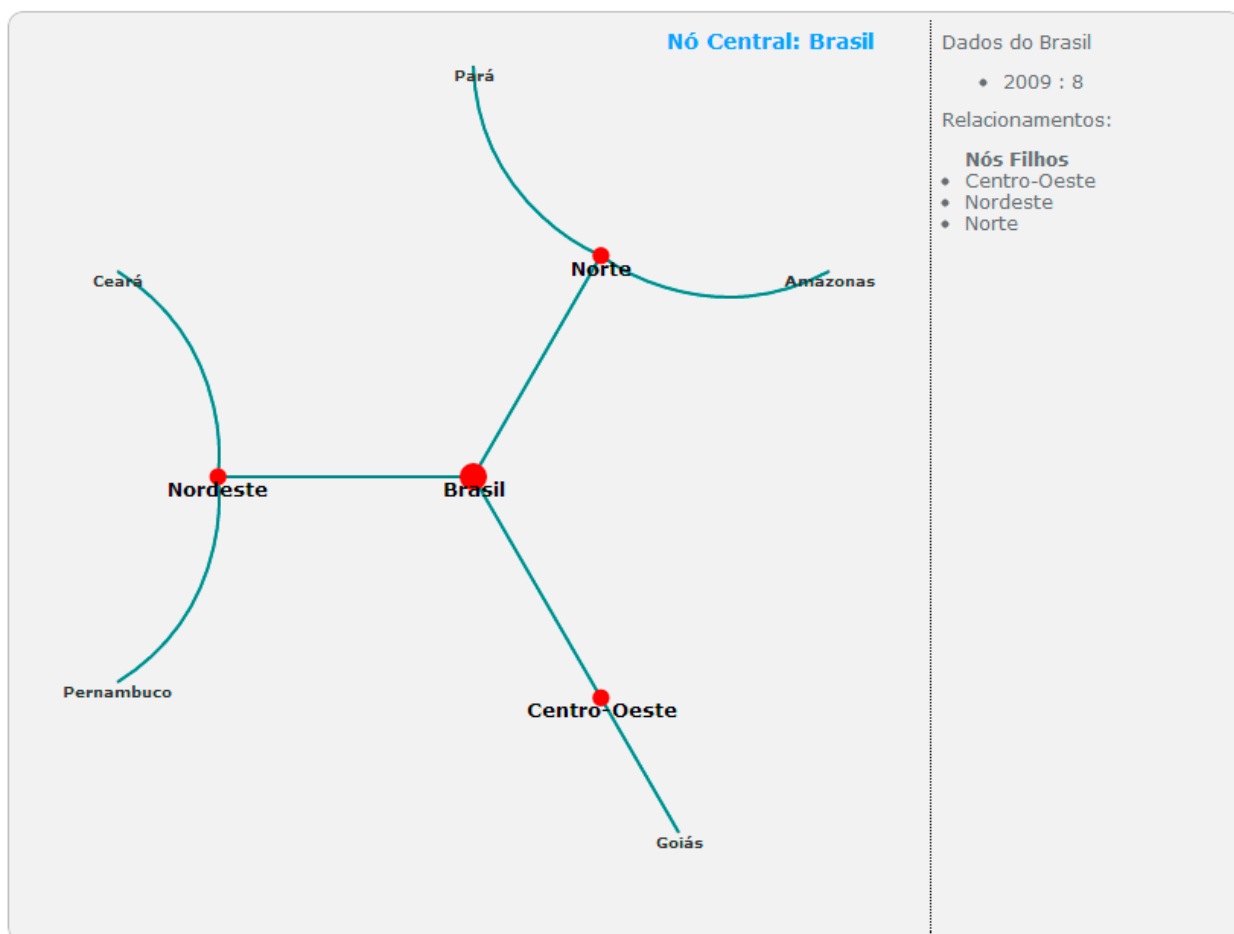


Figura 10 – Visualização gerada pelo XML da Figura 9.