



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO

Marcos Senna Benaion Leal

Aspectos de implementação de mapas choropleth hierárquicos e interativos

Belém – Pará

2017

Marcos Senna Benaion Leal

Aspectos de implementação de mapas choropleth hierárquicos e interativos

Trabalho de Conclusão de Curso apresentado para obtenção do título de Bacharelado em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Faculdade de Computação. Universidade Federal do Pará.

Orientador: Prof. Dr. Carlos Gustavo Resque dos Santos

Coorientador: Prof. Dr. Biachi Serique Meiguins

Belém – Pará

2017

Marcos Senna Benaion Leal

Aspectos de implementação de mapas choropleth hierárquicos e interativos

Trabalho de Conclusão de Curso apresentado para obtenção do título de Bacharelado em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Faculdade de Computação. Universidade Federal do Pará.

Belém – Pará, 22 de Setembro de 2017.

Prof. Dr. Carlos Gustavo Resque dos Santos
Orientador

Prof. Dr. Biachi Serique Meiguins
Coorientador

Prof. Dr. Jefferson Magalhães de Moraes
Membro Interno

Me. Tiago Davi Oliveira de Araújo
Membro Convidado

Belém – Pará
2017

Agradecimentos

Agradeço, primeiramente, a minha família por terem me apoiado nesta caminhada e possibilitarem toda a estrutura para chegar onde estou.

Agradeço ao meu professor orientador Prof. Dr. Carlos Gustavo Resque dos Santos e coorientador Prof. Dr. Biachi Serique Meiguins por seus direcionamentos, confiança e incentivo que me ajudou bastante a concluir este trabalho.

Aos membros do laboratório LabVis da UFPA pelas orientações durante a elaboração deste trabalho, além dos momentos de descontração e troca de idéias durante este período juntos.

Aos meus amigos, pelas alegrias, tristezas e dores compartilhadas. Com vocês, esses longos e duros anos certamente foram mais acolhedores e divertidos.

Resumo

Com o advento de conceitos como *Big Data* e *Smart City* a cidade e sua população têm gerado uma grande quantidade de dados sobre setores da sociedade como: saúde, transporte público, saneamento básico, ensino, entre outros. Contudo, esse volume crescente de dados pouco são relacionados aos seus contextos demográficos, e quando esses dados são representados por mapas, eles de maneira geral não apresentam a interatividade para explorar esses dados, seja alterando o território de estudo, aumentando ou diminuindo sua abrangência, relacionando outros dados para análise, codificando visualmente um dado, entre outras possibilidades. Assim, esse trabalho tem como objetivo o entendimento de como representar dados demográficos, mais especificamente como representar dados em mapas geográficos, e como torná-los interativos para o usuário, apresentando como produto final um software de código livre para visualização de mapas geográficos *choropleth* hierárquicos e interativos para Web. Para isso, são apresentadas a arquitetura, metodologia de implementação, tecnologias utilizadas, aspectos de implementação, e as funcionalidades obtidas com a implementação.

Palavras-chaves: Visualização da Informação, Mapas *Choropleth*, Desenvolvimento *Web*.

Abstract

With the advent of concepts such as Big Data and Smart City the city and its population has generated a lot of data about sectors of society such as health, public transportation, basic sanitation, education, among others. However, this growing volume of data is little related to their demographic contexts, and when these data are represented by maps, they generally do not have the interactivity to explore this data, either by changing the study territory, increasing or decreasing its scope, relating other data for analysis, visually encoding a data, among other possibilities. Thus, this work aims to understand how to represent demographic data, more specifically how to represent data in geographic maps and how to make them interactive for the user, presenting as a final product a open-source software for visualizing hierarchical and interactive Choropleth geographic maps for the Web. To this end, the architecture, implementation methodology, technologies used, implementation aspects and the functionalities obtained are presented.

Key-words: Information Visualization, Choropleth Maps, Web Development.

Lista de ilustrações

Figura 1 – Distribuição Geográfica das bibliotecas públicas dos Estados Unidos.	15
Figura 2 – Mapa exibindo o volume de ligações telefônicas nos Estados Unidos no horário de 6:00 pm (EST).	16
Figura 3 – Visualização das redes de telecomunicações em larga escala utilizando <i>Network Map</i>	17
Figura 4 – <i>Flow Map</i> mostrando a migração de pessoas da Noruega e Letônia para outros lugares.	18
Figura 5 – Visualização em linhas do fluxo de tráfego IP dos nós externos aos nós internos.	19
Figura 6 – Visualização em linhas com a aplicação do <i>Edge Bundling</i>	19
Figura 7 – Mapa <i>Choropleth</i> mostrando a taxa de desemprego nos Estados Unidos, segundo dados do <i>Bureau Labor Statistics</i> de 2016.	20
Figura 8 – Mapa dos Estados Unidos sobre as eleições presidenciais de 2000.	21
Figura 9 – Cartograma mostrando a população (correspondente ao tamanho) e percentagem de votos (correspondente a cor) da eleição presidencial de 2000 nos Estados Unidos.	21
Figura 10 – Variáveis visuais, suas propriedades visuais no contexto dos pontos, linhas ou áreas e modos de implantação em mapas.	23
Figura 11 – Página da <i>Web</i> do IBGE Cidades mostrando frota de veículos no Pará.	24
Figura 12 – Página da <i>Web</i> do IBGE Cidades mostrando frota de veículos no Pará por tipo durante 2010-2016.	25
Figura 13 – Página <i>Web</i> do GeoCAPES mostrando a distribuição da Concessão de bolsas de Pós-Graduação no Brasil.	26
Figura 14 – Página <i>Web</i> do GeoCAPES mostrando a distribuição da Concessão de bolsas de Pós-Graduação nos municípios do Pará.	27
Figura 15 – <i>Pipeline</i> de visualização.	28
Figura 16 – Representação da arquitetura da aplicação, camadas pertencentes e seus componentes.	30
Figura 17 – Diagrama de Casos de Uso da Ferramenta de Visualização.	32
Figura 18 – Diagrama de Classes da Ferramenta de Visualização.	33
Figura 19 – Diagrama de Sequência da funcionalidade de Exibição de Mapas.	35
Figura 20 – Diagrama de Sequência da funcionalidade "Alterar Filtro".	36
Figura 21 – Diagrama de Sequência da funcionalidade "Detalhes sobre demanda".	37
Figura 22 – Arquitetura da ferramenta com as tecnologias utilizadas em cada componente.	40

Figura 23 – Interface do <i>geojson.io</i> durante a adição de um polígono representando um bairro de Belém.	41
Figura 24 – Bairros do município de Belém.	42
Figura 25 – Trecho da página Guia do Eleitor descrevendo os limites do bairro Cidade Velha.	42
Figura 26 – Definindo valores para os campos nome, id (número de identificação), ponto latitude e longitude para cada bairro de Belém.	43
Figura 27 – Mapa do Brasil pronto para ser exportado em um arquivo <i>TopoJSON</i>	44
Figura 28 – Planilha contendo população e 'id' dos estados brasileiros.	45
Figura 29 – Lista de cores RGB com tom verde, partindo do verde claro ao verde escuro, respectivamente de baixo para cima.	47
Figura 30 – Exemplo de criação de uma textura utilizando a biblioteca <i>Textures.js</i>	47
Figura 31 – Visão geral do protótipo.	48
Figura 32 – Mapa do Brasil.	49
Figura 33 – Mapa do estado do Pará (após o clique no estado do Pará no mapa do Brasil).	49
Figura 34 – Mapa dos bairros do município de Belém.	50
Figura 35 – Hierarquia percorrida até chegar ao mapa dos bairros de Belém.	50
Figura 36 – Dados exibidos sobre o bairro Aeroporto.	51
Figura 37 – Dados exibidos sobre o bairro São Brás.	51
Figura 38 – Ampliação do mapa dos municípios do Pará, com destaque para Ananindeua.	52
Figura 39 – Mapa de Belém com foco na área de Mosqueiro.	53
Figura 40 – Mapa do Brasil sem filtros ativados.	54
Figura 41 – Mapa do Brasil com um filtro ativado.	54
Figura 42 – Mapa do Brasil com dois filtros ativados simultâneamente.	55
Figura 43 – Exemplo de mudança do tema de cores para verde.	56
Figura 44 – Exemplo de modificação das texturas, mesclando diferentes formas geométricas.	56

Lista de abreviaturas e siglas

CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
D3	Data-Driven Documents
GeoJSON	Geospatial Data JavaScript Object Notation
GIS	Geographic Information System
GViz	Visualização Geográfica
HTML	HyperText Markup Language
IBGE	Instituto Brasileiro de Geografia e Estatística
ID	Identificador
IP	Internet Protocol
JSON	JavaScript Object Notation
MVC	Model-View-Controller
SVG	Scalable Vector Graphics
TopoJSON	Topological Data JavaScript Object Notation
TSV	Tab Separated Values
XML	eXtensible Markup Language

Sumário

1	Introdução	10
1.1	Justificativa	10
1.2	Objetivos	11
1.2.1	Objetivo geral	11
1.2.2	Objetivos específicos	11
1.2.3	Estrutura do Trabalho	11
2	Referencial teórico	12
2.1	Visualização da Informação	12
2.2	Visualização Geográfica	13
2.2.1	Visualização de dados pontuais	14
2.2.2	Visualização de dados lineares	16
2.2.3	Visualização de dados de área	20
2.2.4	Variáveis visuais em mapas	22
2.3	Aplicações Relacionadas	24
3	Desenvolvimento do Protótipo	28
3.1	Arquitetura	28
3.2	Modelagem de software	31
3.3	Tecnologias Utilizadas	37
3.4	Aspectos de Implementação	40
3.4.1	Obtenção de dados	40
3.4.2	Tratamento e análise dos dados	43
3.4.3	Limitações das pesquisas	45
3.5	Funcionalidades	46
3.6	Protótipo	47
4	Considerações Finais	57
4.1	Trabalhos futuros	57
	Referências	59

1 Introdução

Os dados obtidos sobre uma cidade podem informar a realidade de seus habitantes através de estatísticas sociais, números da economia e relatórios de infra-estrutura, sendo esses apenas um pequeno exemplo das informações que podem ser coletadas. Contudo, apenas reunir os dados não fornece conhecimento suficiente para realizar inferências sobre os mesmos, ou responder perguntas como: "O que pode ser melhorado em transportes públicos para atender a população crescente?". A análise de dados pode ajudar a entender relações entre diferentes fenômenos, observar padrões recorrentes e até mesmo gerar novos dados.

As visualizações de mapas são interessantes para mostrar fenômenos georreferenciados, ou seja, que estejam relacionados a regiões específicas do mundo real. Os mapas também são intuitivos, permitindo que uma pessoa associe rapidamente dados aos mapas através de recursos visuais, como por exemplo cores, pontos, texturas, etc.

Este trabalho propõe a criação de uma arquitetura e implementação de um sistema de visualização de mapas, utilizando representação *Choropleth*, hierarquias de navegação entre mapas e interações através de filtros, coordenação entre gráficos e operações de *pan* (transladar mapa) e *zoom* (ampliar/reduzir elementos do mapa).

Para a demonstração do protótipo da ferramenta foram obtidos e utilizados os dados geográficos, de área e demografia dos estados do Brasil, municípios do Pará e bairros de Belém; sendo o último (bairros de Belém) criado pelo autor. Estas informações são exibidas no protótipo com o auxílio de tecnologias *Web*, como por exemplo *JavaScript*, e apresentadas no mapa utilizando recursos visuais de cor e textura.

1.1 Justificativa

Ao acessar os dados estatísticos e o mapa dos bairros de Belém (através do site da Prefeitura de Belém) observou-se uma dissociação entre esses dados no sentido de que não há uma relação entre as informações em um ou mais tipos de visualização (gráfico de barras, setores, mapas, etc.). A criação de um mapa interativo pode se mostrar benéfico para o município orientando projetos de transporte, mobilidade urbana e políticas de segurança pública. Além disso, o mapa pode simplificar as informações do anuário (ou outra fonte de dados estatísticos) em uma única visualização (compondo mapas, gráficos de barras, gráficos de setores, dentre outros tipos), tornando-a mais fácil e rápida de interpretar, encontrar informações e tomar decisões. Tal oportunidade gera uma demanda de criar uma visualização de mapas que apresente estes dados de forma a mostrar os dados

à sua região, por exemplo mostrar a quantidade de focos de queimadas por município.

As características recomendáveis para uma boa ferramenta desse tipo são: permitir a utilização de múltiplas variáveis visuais, tornando-a mais intuitiva possível; possuir interações com o usuário, como por exemplo remover elementos da visualização ou exibir detalhes sob demanda ao selecionar os itens (através de um outro gráfico, por exemplo) e possuir uma hierarquia de mapas que possibilite a exibição de outros mapas ao clicar nos elementos, além de poder navegar pela hierarquia.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo geral deste trabalho consiste em desenvolver uma arquitetura e realizar uma implementação de mapas hierárquicos interativos utilizando visualização *choropleth* em ambiente Web.

1.2.2 Objetivos específicos

- Analisar as variáveis visuais possíveis de serem aplicadas aos dados;
- Definir técnicas de categorização de dados;
- Escolher interações adequadas pertinentes ao nível hierárquico do mapa;
- Definir a arquitetura da ferramenta;
- Modelar a arquitetura e os aspectos de implementação;
- Avaliar as tecnologias que servirão de base para a visualização;
- Implementar o protótipo;

1.2.3 Estrutura do Trabalho

O trabalho está estruturado da seguinte forma, no Capítulo 1 é delimitado o problema com relação aos mapas e descrito os objetivos da pesquisa. No Capítulo 2 são informados os conceitos e outras implementações de visualizações de mapas que participam na construção deste trabalho. No Capítulo 3 é detalhado o processo de criação do protótipo da ferramenta de visualização, a metodologia de pesquisa e aquisição dos dados, descrição da arquitetura, as tecnologias utilizadas e outros aspectos de implementação. Por fim no Capítulo 4 são expostas as considerações finais sobre a pesquisa e os trabalhos futuros que podem ser originados com ela.

2 Referencial teórico

Neste capítulo serão abordados brevemente os conteúdos pesquisados para elaboração do trabalho, conceitos sobre a comunicação de informações através de mapas, quais os elementos principais a serem considerados na criação de mapas *Choropleth* e a hierarquia aplicada aos mapas.

Para a elaboração deste trabalho foram pesquisados conteúdos acadêmicos relacionados à visualização da informação, visualização geográfica e tecnologias *Web*.

2.1 Visualização da Informação

Segundo (SPENCE, 2007), a visualização é uma atividade cognitiva humana em que a pessoa formula um modelo ou uma imagem mental de algo. Essa "imagem" pode ser transmitida através de estímulos sensoriais tais como sons, percepções táteis e visuais, esta última é a mais comum de se encontrar.

Escolher a representação adequada, com o auxílio de técnicas de computação gráfica, para auxiliar a compreensão e análise de um determinado conjunto de dados (ou de um problema, a fim de tornar sua solução transparente) é a tarefa principal da Visualização da Informação (SPENCE, 2007). Além disso em toda visualização são utilizadas variáveis visuais. Estas são definidas como os diferentes estímulos visuais que a retina pode perceber em uma visualização.

Segundo (SHNEIDERMAN, 1996) as tarefas que uma visualização deve proporcionar ao usuário incluem:

- Visão Geral (*Overview*) - Permite ter uma visão geral sobre a coleção de dados completa.
- *Zoom* - Permitir que sejam focados os elementos de interesse.
- Filtro - Consiste em filtrar os itens que não são de interesse.
- Detalhes sobre demanda - Ao selecionar um item ou um grupo de elementos, exibir mais informações.
- Relacionar - Observar as relações entre os elementos.
- Histórico - Manter um histórico de ações pra desfazer, refazer e proporcionar um refinamento progressivo.

- Extração - Permitir a extração de subconjuntos de dados e dos parâmetros de consultas à eles.

Ainda de acordo com (SHNEIDERMAN, 1996), as visualizações podem ser classificadas de acordo com o tipo de dado utilizado:

- Unidimensional - São dados lineares, dispostos de maneira sequencial e ordenados por uma única característica. Podem ser documentos textuais, como um texto, código fonte de um programa ou uma lista de caracteres em ordem alfabética.
- Bidimensional - Dados com duas dimensões apresentam informações sobre um espaço, como por exemplo plantas de casas, a organização dos textos/figuras de um jornal ou mapas geográficos. Em resumo, cada item ocupa uma parte do espaço e detalha informações sobre este.
- Tridimensional - São dados com três dimensões e compostos pelos objetos do mundo real (moléculas, construções) que incluem características de superfície e volume.
- Temporal - São dados que representam tempo, mostrando um histórico sobre um evento específico (registros médicos de um paciente, o processo de desenvolvimento de um projeto de software). Embora lineares, se diferenciam dos dados unidimensionais devido possuírem um ponto de início e fim e também permitir a sobreposição dos dados.
- Multidimensional - Dados multidimensionais demonstram relações entre itens, identificação de padrões e comportamento dos dados. Esses itens possuem diversos atributos e cada atributo pode ser representado em uma dimensão.
- Árvores - São coleções de itens que possuem relações de filho e pai. Devido a esta relação, apresentam níveis em que itens podem ser irmãos entre si ou um item possuir várias ramificações. Hierarquias e o acesso a um diretório de arquivos de um computador são exemplos de árvores.
- Redes - Demonstram relações entre itens mas com ênfase no custo entre essas relações (ou caminhos).

O tipo de visualização que será abordado neste trabalho é a visualização geográfica bidimensional.

2.2 Visualização Geográfica

De acordo com (RAMOS, 2003) a Visualização Geográfica é definida como uma forma de visualização de informações utilizando mapas, com o objetivo de avaliar e de-

envolver métodos visuais que auxiliam a análise, exploração, síntese e apresentação de informações georreferenciadas. Os dados geoespaciais, os quais irão ser utilizados na visualização geográfica, são diferentes dos outros tipos de dados, pois nesses são descritos objetos ou fenômenos relacionados a localizações específicas do mundo real (WARD; GRINSTEIN; KEIM, 2010).

Segundo (WARD; GRINSTEIN; KEIM, 2010) os mapas representam o mundo através de pontos, linhas ou áreas e especifica cada uma dessas representações a um tipo de fenômeno específico:

- Pontual - Não possuem extensão, podem ser considerados zero-dimensionais e serem especificados através das coordenadas de latitude e longitude. Exemplos de dados pontuais são construções, agregações mensuradas e cidades.
- Linear - Possuem comprimento, mas essencialmente sem largura. São considerados unidimensionais e podem ser especificados através de um par de coordenadas de latitude e longitude. Exemplos de dados em linha são ruas, malhas das redes de telecomunicação e divisões entre países.
- Areal - São bidimensionais, possuem largura e comprimento e são formados por pares de coordenadas de latitude e longitude que fecham e delimitam uma região. Alguns exemplos de dados de área são lagos e unidades político-administrativas (por exemplo cidades e estados).

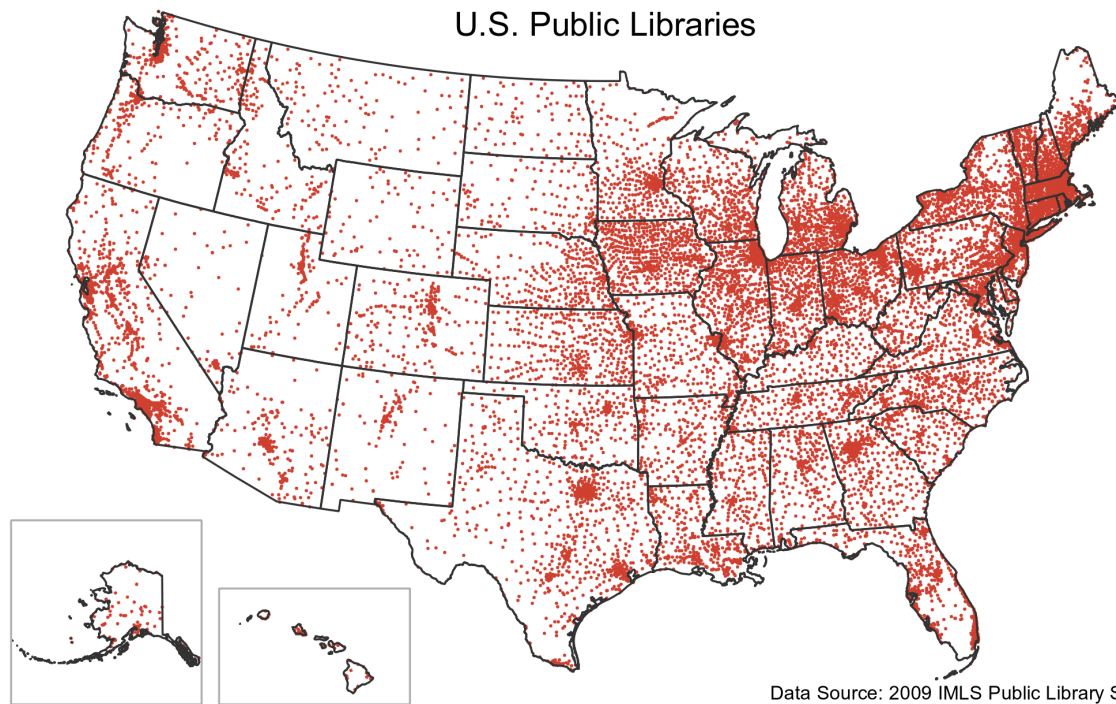
Uma breve introdução a estes tipos de visualizações (ou representações) serão realizadas nas próximas subseções.

2.2.1 Visualização de dados pontuais

Dados pontuais são tipos de dados espaciais de natureza discreta, mas que podem ser utilizados também para representar fenômenos contínuos, como a medição de temperaturas de determinados lugares (WARD; GRINSTEIN; KEIM, 2010). As possíveis visualizações de mapas com dados pontuais são *Dot maps* (mapa com pontos) e *Pixel maps*.

Os *Dot Maps* representam os fenômenos pontuais através da aplicação de um símbolo ou pixel no local em que cada fenômeno ocorre. Outros parâmetros podem ser mapeados aos símbolos, como por exemplo o tamanho e a cor do mesmo. Um símbolo comum de ser utilizado em *dot maps* é o círculo, entretanto qualquer símbolo pode ser aplicado (quadrados, barras, etc) (WARD; GRINSTEIN; KEIM, 2010). A Figura 1 mostra um exemplo de um *dot map*.

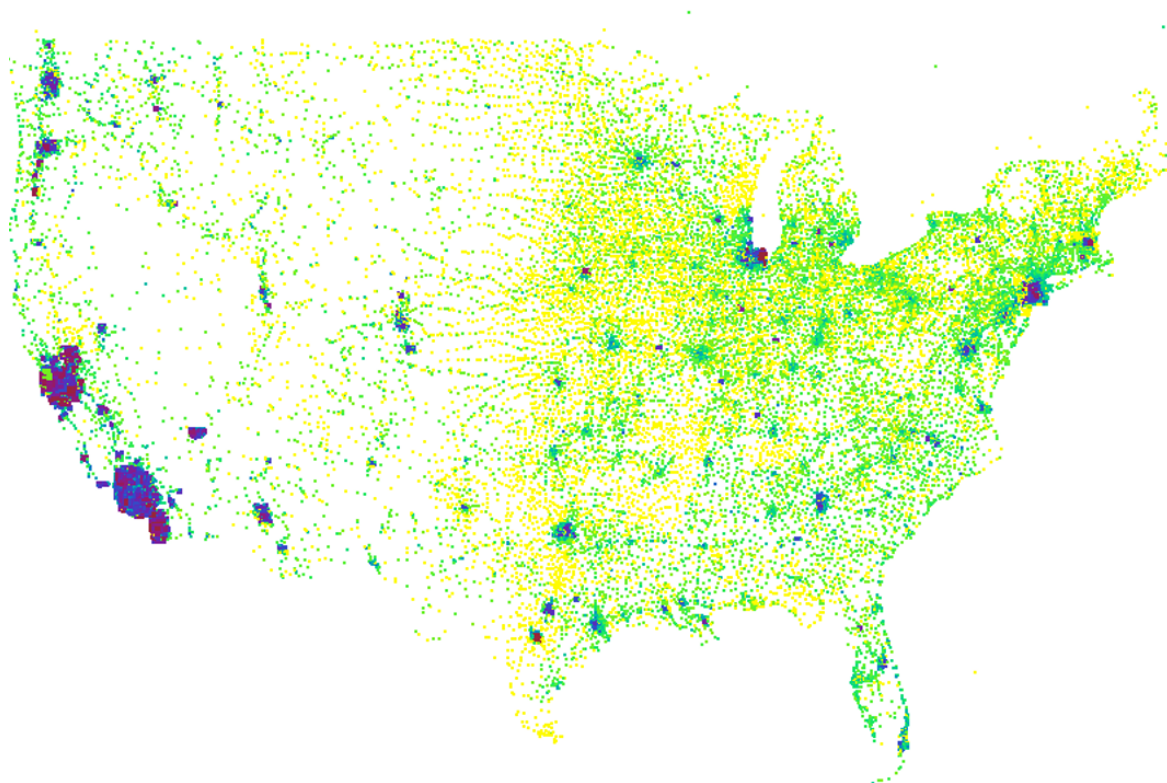
Figura 1 – Distribuição Geográfica das bibliotecas públicas dos Estados Unidos.



Fonte – IMLS - Institute of Museum and Library Services - Pesquisa realizada em 2009 das bibliotecas públicas dos Estados Unidos

Um problema que ocorre nos *dot maps* em conjuntos de dados grandes é a sobreposição dos pontos de dados em áreas super populadas, enquanto que as áreas com pouca população ficam praticamente vazias. Com isso uma abordagem que evita sobreposição é o *Pixel Map*, o qual reposiciona os *pixels* (pequenos pontos coloridos na tela) que iriam sobrepor outros em subregiões próximas, definidas através de um algoritmo de reposicionamento e da partição das subregiões nos dados (WARD; GRINSTEIN; KEIM, 2010). A Figura 2 mostra um exemplo de um *Pixel Map*.

Figura 2 – Mapa exibindo o volume de ligações telefônicas nos Estados Unidos no horário de 6:00 pm (EST).



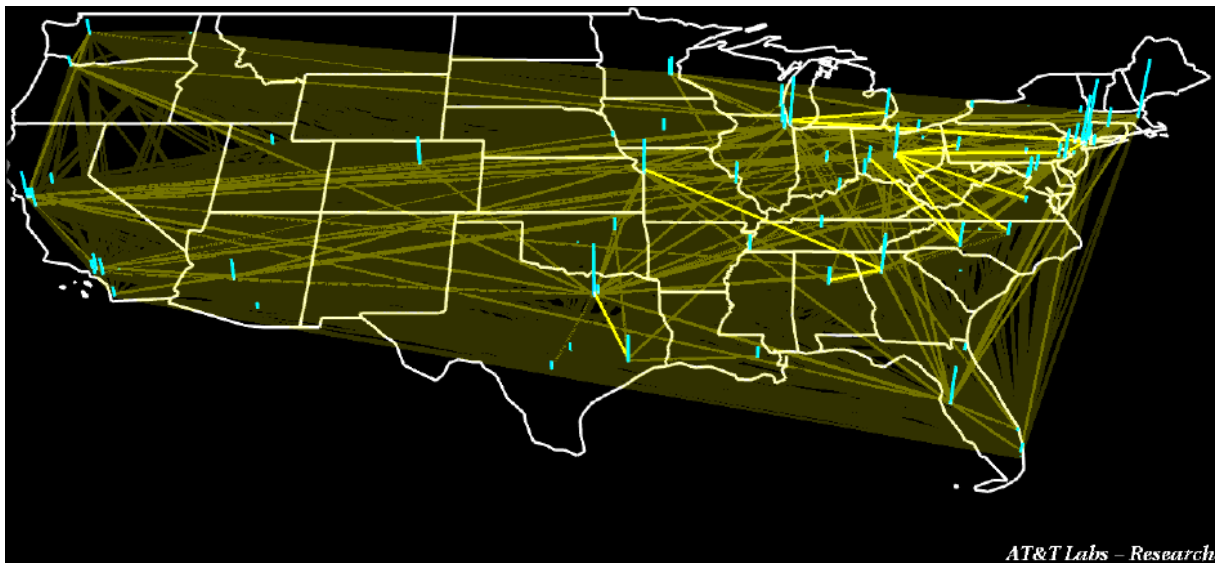
Fonte – (WARD; GRINSTEIN; KEIM, 2010)

2.2.2 Visualização de dados lineares

Os dados que representam fenômenos lineares são descritos pelos pares de pontos entre as extremidades das linhas, as quais são especificadas pela latitude e longitude dos pontos. Os parâmetros que podem ser mapeados nas linhas são a largura, textura, cor e uma legenda. Além disso os nós nas extremidades da linha também podem ser mapeados em variáveis visuais como por exemplo cor, forma, tamanho e rótulo. As linhas não necessariamente precisam estar retas, onde dependendo da tarefa ou aplicação dos dados podem ser utilizadas polilinhas ou curvas (WARD; GRINSTEIN; KEIM, 2010).

Os *Network Maps* utilizam as representações lineares para mostrar a conectividade de redes para entender a sua estrutura geral e comportamento. As variáveis visuais de cor e forma são utilizadas para codificar informações sobre o nó, enquanto que a cor e tamanho da linha são utilizados para exibir informações sobre as ligações (*links*) entre os nós. Um problema aparente nessas visualizações de grandes redes (com muitas conexões) em mapas é a sobreposição de linhas em áreas densas (WARD; GRINSTEIN; KEIM, 2010). A Figura 3 mostra um exemplo de um *Network Map*.

Figura 3 – Visualização das redes de telecomunicações em larga escala utilizando *Network Map*.

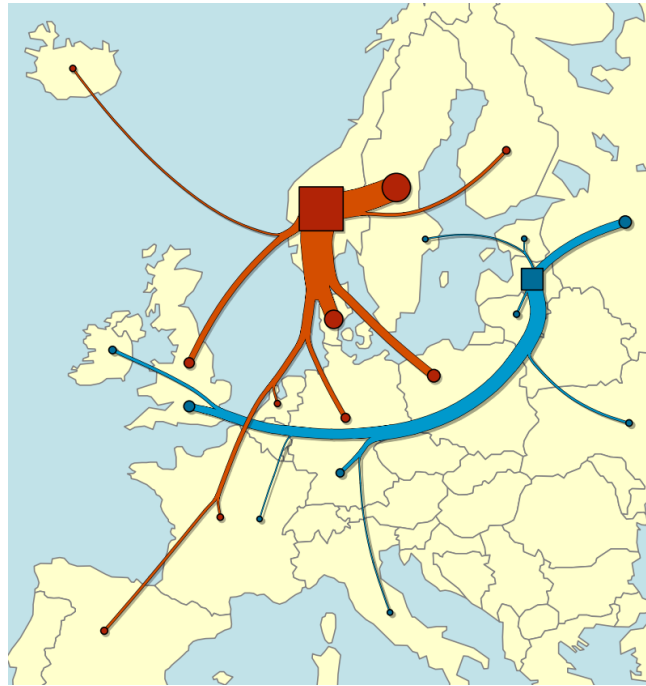


Fonte – (WARD; GRINSTEIN; KEIM, 2010)

Para amenizar o problema de sobreposição em *Network maps* as técnicas de *Flow Map* (mapa de fluxo) e *Edge Bundling* (agrupamento de bordas) são empregadas.

A técnica de *Flow Map* tenta minimizar o cruzamento de bordas e as distorções causadas pela posição dos nós. Nesta situação um algoritmo de clusterização hierárquica é aplicado levando em conta as posições dos nós e os fluxos entre eles, com o cuidado de manter ainda a posição relativa entre eles. Desta forma, os fluxos são realocados e agrupados (e os nós reposicionados) com a finalidade de reduzir a poluição visual causada pelo excesso de linhas na visualização (WARD; GRINSTEIN; KEIM, 2010). A Figura 4 mostra um exemplo de um *Flow Map*.

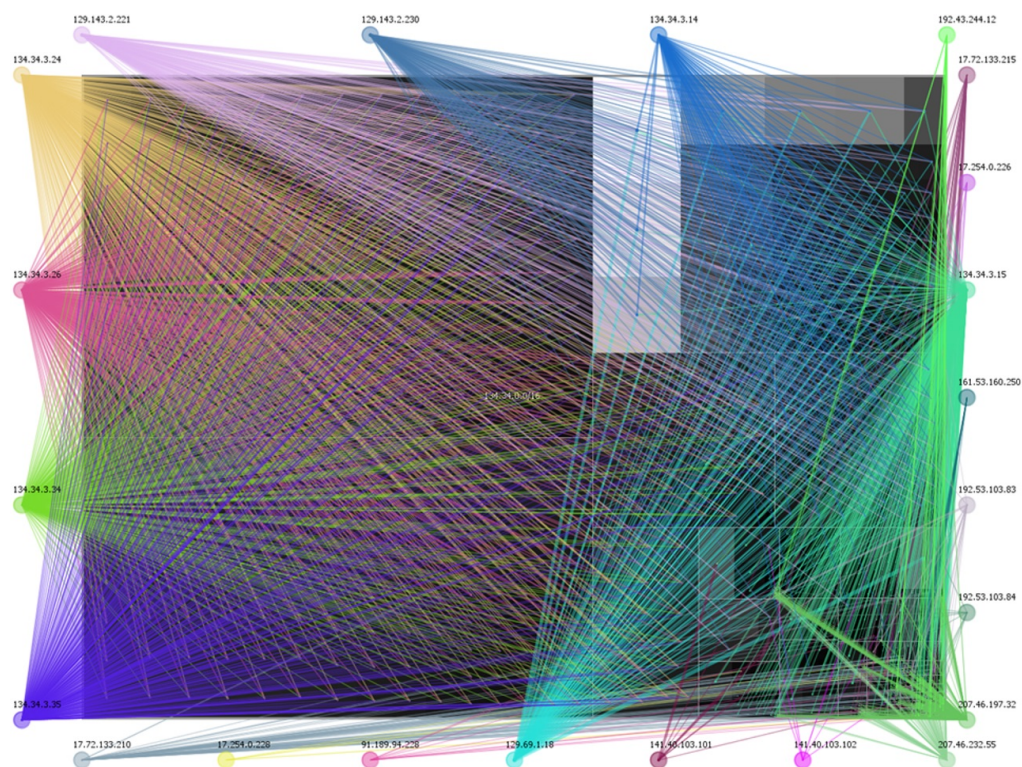
Figura 4 – *Flow Map* mostrando a migração de pessoas da Noruega e Letônia para outros lugares.



Fonte – (BUCHIN; SPECKMANN; VERBEEK, 2011)

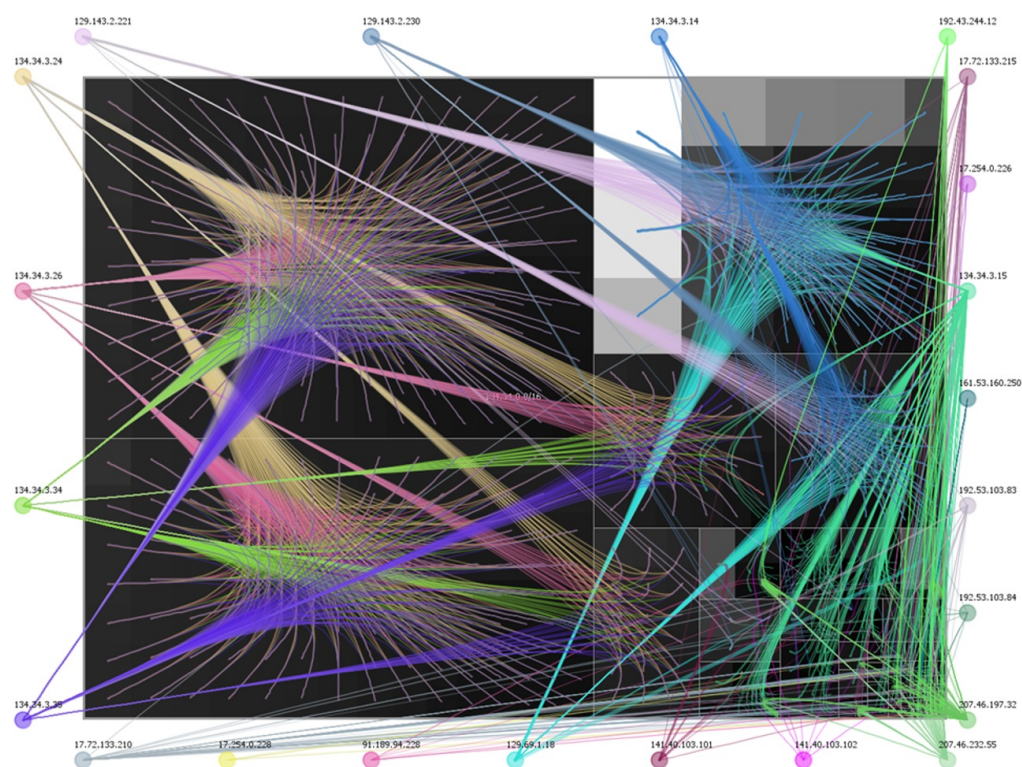
Outra técnica utilizada é o *Edge Bundling* na qual as linhas são curvadas e aproximadas umas das outras (WARD; GRINSTEIN; KEIM, 2010). As Figuras 5 e 6 mostram, respectivamente, antes e depois da aplicação do *Edge Bundling* em uma visualização sobre o fluxo de tráfego IP em redes de larga escala.

Figura 5 – Visualização em linhas do fluxo de tráfego IP dos nós externos aos nós internos.



Fonte – (WARD; GRINSTEIN; KEIM, 2010)

Figura 6 – Visualização em linhas com a aplicação do *Edge Bundling*.



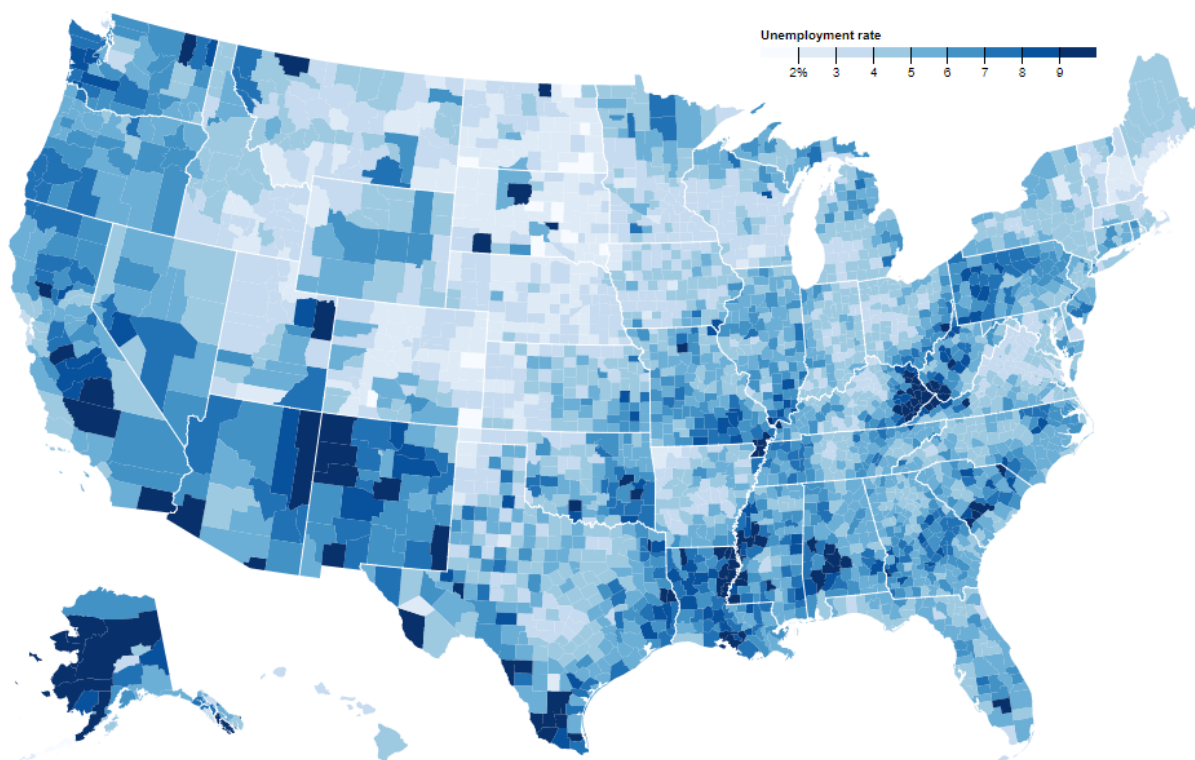
Fonte – (WARD; GRINSTEIN; KEIM, 2010)

2.2.3 Visualização de dados de área

Os dados de área são aqueles que representam áreas ou regiões do mapa e seus valores agregados (de acordo com o dado de estudo). Para visualizar fenômenos de área são utilizados mapas temáticos, dentre eles a representação *Choropleth* e cartogramas.

O nome *Choropleth* tem sua origem grega, onde *choro* significa área e *pleth* significa valor, podendo ser chamado em português de coroplético(a) ou coropleto. Nesse tipo de mapa os valores estatísticos são codificados nas regiões do mapa, as quais são compostas por polígonos fechados (formando a área), e nelas há uma coloração de acordo com o valor do dado. Os mapas *Choropleth* são utilizados para enfatizar a distribuição espacial de um ou mais atributos geográficos e realçar padrões desses valores (WARD; GRINSTEIN; KEIM, 2010). A Figura 7 mostra um exemplo de um mapa *Choropleth*.

Figura 7 – Mapa *Choropleth* mostrando a taxa de desemprego nos Estados Unidos, segundo dados do *Bureau Labor Statistics* de 2016.



Fonte – Mike Bostock - *Choropleth Maps with D3*

Cartogramas são tipos de mapas geográficos que distorcem a geografia da visualização de acordo com os valores estatísticos, aumentando ou diminuindo o tamanho das áreas. Essa distorção pode implicar na perda da percepção da forma do mapa original ou de sua topologia. As Figuras 8 e 9 mostram o mapa original e a sua versão em cartograma, respectivamente, sobre as eleições presidenciais de 2000 dos Estados Unidos. As cores nos mapas representam quais candidatos (*George W. Bush* em Azul e *Al Gore* em

tom avermelhado) venceram nos estados.

Figura 8 – Mapa dos Estados Unidos sobre as eleições presidenciais de 2000.

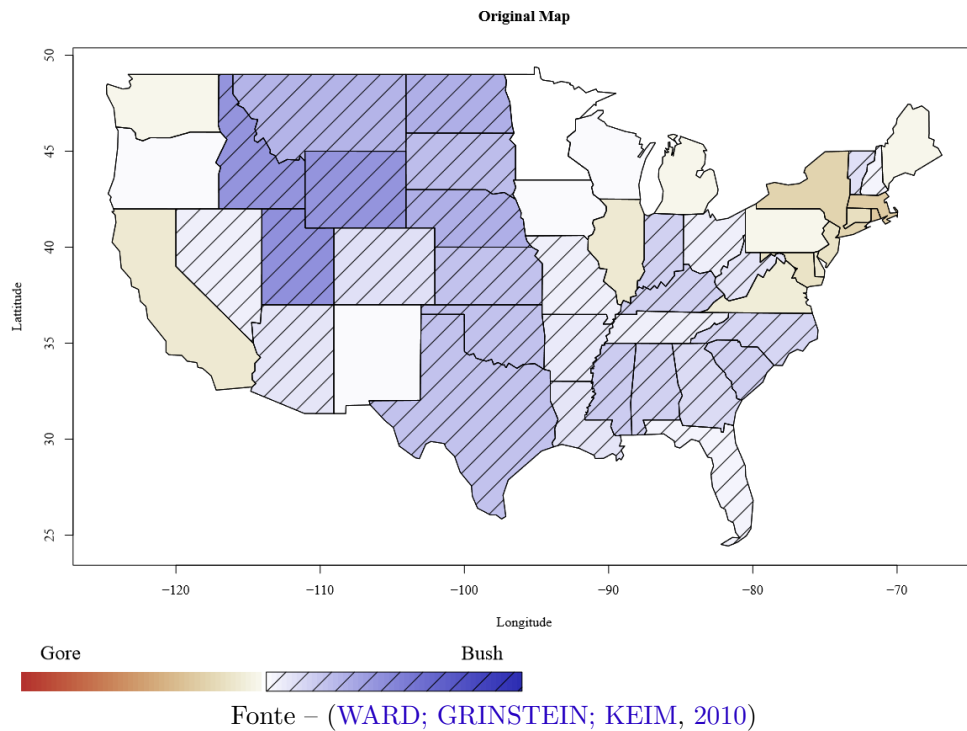
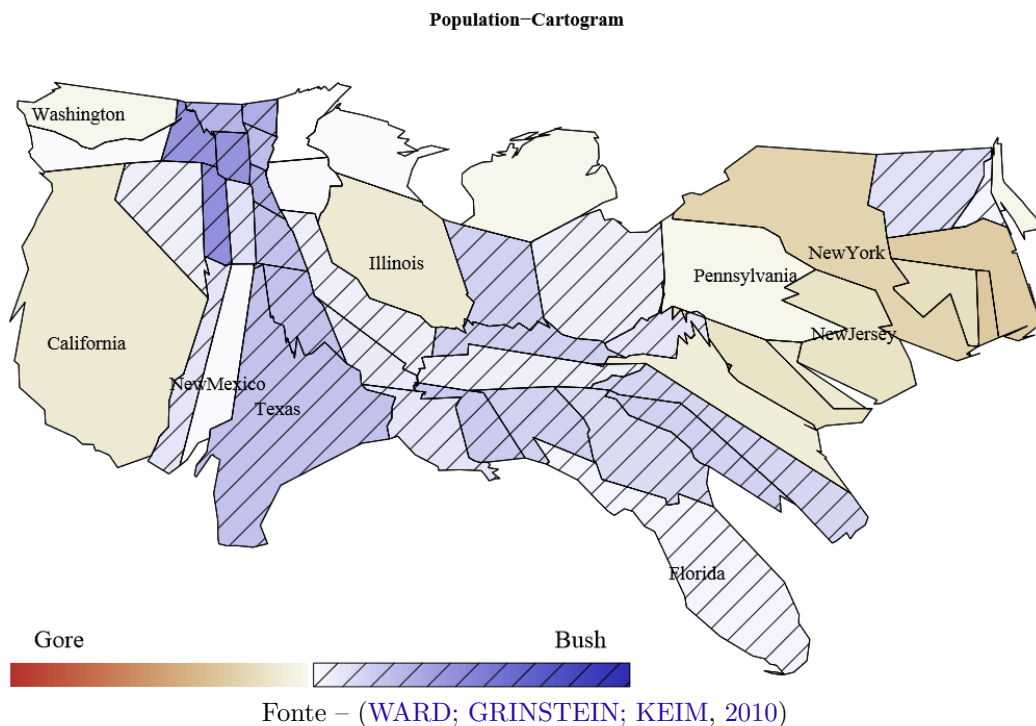


Figura 9 – Cartograma mostrando a população (correspondente ao tamanho) e porcentagem de votos (correspondente a cor) da eleição presidencial de 2000 nos Estados Unidos.



2.2.4 Variáveis visuais em mapas

Mapas são utilizados de muitas maneiras, por exemplo, para fornecer informações sobre uma localização em particular, fornecer informação geral sobre padrões espaciais ou comparar padrões em múltiplos mapas. O mapeamento das propriedades dos dados espaciais para as variáveis visuais devem refletir no objetivo de análise que se quer alcançar (WARD; GRINSTEIN; KEIM, 2010). Para representar as informações espaciais de população e área no mapa são utilizadas variáveis visuais. As variáveis visuais para dados espaciais são:

- Tamanho - Determina o tamanho que o elemento irá assumir no mapa (tamanho dos símbolos ou largura das linhas). Pode ser fixo ou variar de acordo com um valor quantitativo.
- Forma - Formas podem ser aplicadas em símbolos ou texturas para significar algo em um atributo categórico.
- Brilho - O brilho pode variar em símbolos, linhas ou áreas para demonstrar um atributo quantitativo e inclusive pode ser combinado com matiz em mapas.
- Cor - Permite distinguir os diferentes elementos no mapa e perceber padrões espaciais quando estão agrupados.
- Orientação - Permite estabelecer uma direção e sentido para os símbolos e/ou texturas.
- Textura - As texturas são combinadas com formas e orientação para possuir um significado. Podem ser aplicadas em conjunto com as cores para adicionar mais um grau de informação.
- Perspectiva de altura - Adiciona uma perspectiva tridimensional para o fenômeno, mapeando os dados com a altura e destacando os pontos, linhas ou áreas na visualização.
- Arranjo de Padrões - Compreende a disposição dos padrões nas texturas representando fenômenos pontuais (padrões nos símbolos), lineares (padrões nas linhas, por exemplo traços e pontos intercalados) ou de área (padrões regulares ou aleatórios das formas).

A Figura 10 mostra um exemplo dessas variáveis visuais organizadas de acordo com o tipo de dado espacial trabalhado e mostrando um exemplo de implantação no mapa.

Figura 10 – Variáveis visuais, suas propriedades visuais no contexto dos pontos, linhas ou áreas e modos de implantação em mapas.

	Tamanho	Forma	Brilho	Cor	Orientação	Espaçamento	Perspectiva de altura	Arranjo de Padrões
Pontual								
Linear								
Areal								

Fonte – Imagem adaptada de (WARD; GRINSTEIN; KEIM, 2010)

Neste trabalho serão utilizadas as variáveis visuais de cor (uma combinação de intensidade e cor, resultando em matiz e brilho), tamanho e textura, onde textura irá combinar orientação e forma.

Devido aos mapas obtidos possuírem fenômenos (população e área) que ocorrem segundo limites político-administrativos (estados, municípios, bairros) a representação mais adequada, além de ser uma visualização simples de interpretar dados, é a representação *Choropleth*. Com isso a representação estipula diferentes valores para cada área. Esses valores passam, primeiramente, por um processo de classificação e as classes resultantes são exibidas nos polígonos relacionados (RAMOS, 2003). Para o processo de classificação dos dados utilizados neste trabalho foi aplicado nestes o algoritmo *Jenks Natural Breaks*.

Segundo (MACWRIGHT, 2013), o algoritmo *Jenks Natural Breaks* busca criar classes maximizando a distância entre as médias das classes, ao mesmo tempo fazendo com que os elementos pertencentes a cada uma das classes tenham um desvio pequeno para o valor da média da classe (ou seja, possuam valores próximos as médias das classes). Essencialmente agrupa os elementos semelhantes e separa os com grandes intervalos entre valores. Com isso, o algoritmo transforma os valores numéricos e contínuos em valores categóricos, mostrando a qual classe cada valor pertence.

Por fim uma interação adicionada é a mudança de hierarquia, na qual a aplicação é estruturada da seguinte forma: o mapa do Brasil possui a maior hierarquia e através dele é possível acessar o mapa dos estados brasileiros, o mapa dos estados mostra os municípios do mesmo e fornece acesso aos bairros, e por último a menor hierarquia é o mapa dos bairros, onde não é possível avançar mais na hierarquia e exibe informações dos bairros

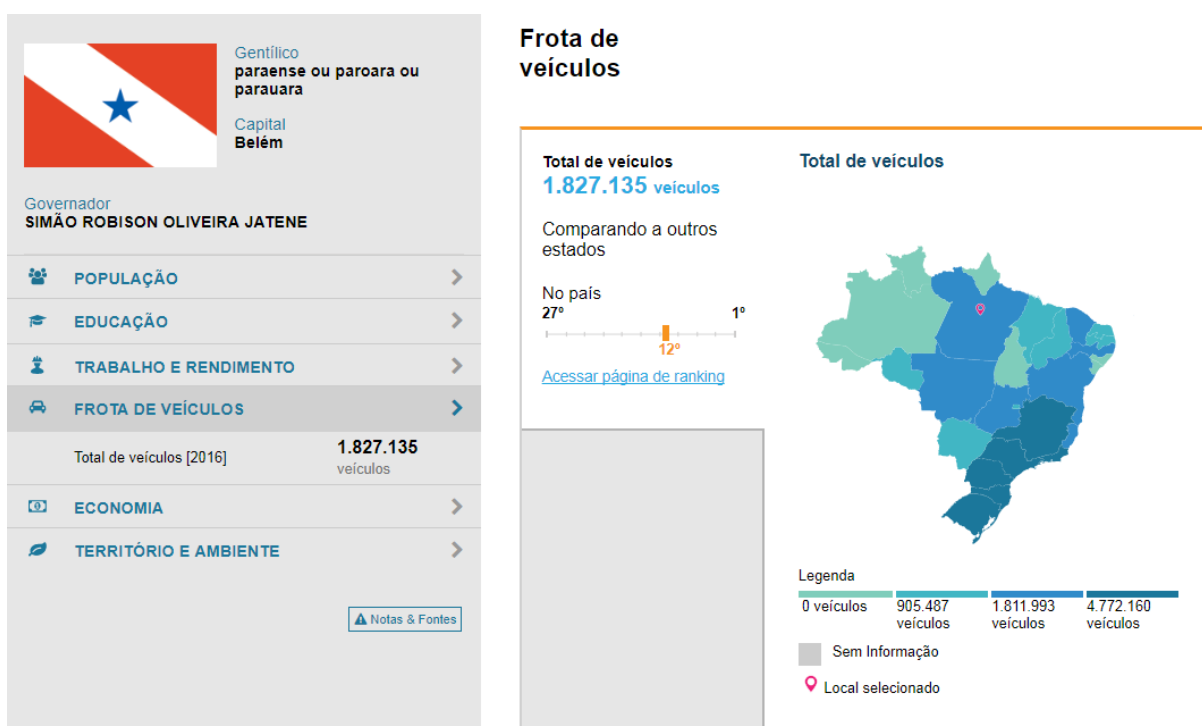
em um gráfico de barras.

2.3 Aplicações Relacionadas

Um bom ponto de partida para analisar funcionalidades interessantes que podem se adequar a ferramenta de visualização a ser criada é o *site* do IBGE Cidades (versão de 2017)¹. A Figura 11 mostra o mapa do Brasil com o tema "Frota de Veículos", ao lado esquerdo da janela do mapa a quantidade de veículos no Pará e um mapa comparativo entre a quantidade de veículos dos estados, com a cor tema azul; onde azul claro representa uma quantidade pequena de carros, na qual aumenta a intensidade até a tonalidade azul escuro, significando uma grande quantidade de carros.

Além disso, na Figura 12, um detalhamento das informações de "Frota de Veículos" do Pará é exibido, fornecendo dados sobre a quantidade de veículos por tipo (Automóvel, Caminhão, Ônibus e Motocicleta) durante o período de 2010 até 2016.

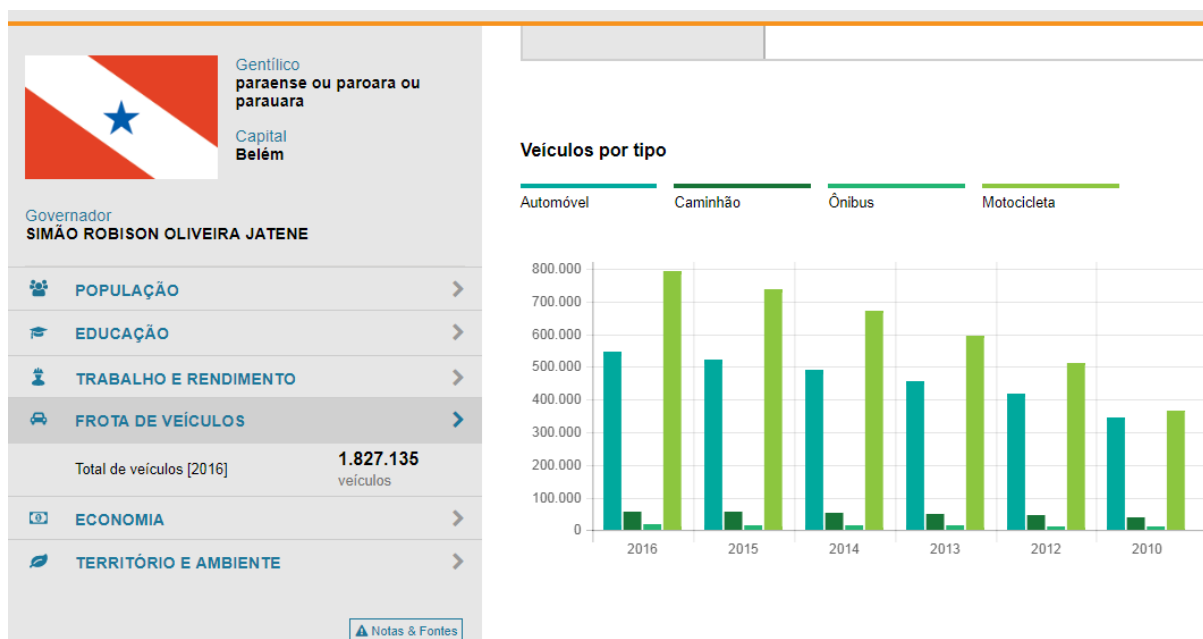
Figura 11 – Página da *Web* do IBGE Cidades mostrando frota de veículos no Pará.



Fonte – Captura de tela do *Site* IBGE Cidades

¹ <https://cidades.ibge.gov.br/>

Figura 12 – Página da Web do IBGE Cidades mostrando frota de veículos no Pará por tipo durante 2010-2016.



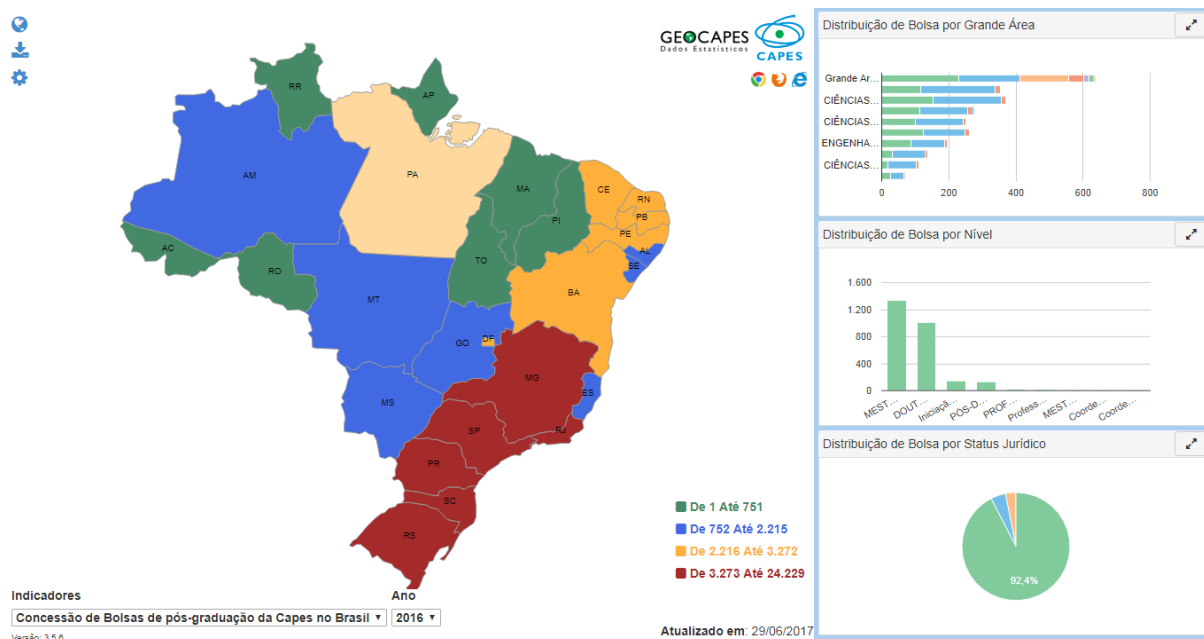
Fonte – Captura de tela do Site IBGE Cidades

Percebe-se uma falta de coordenação entre as informações da Figura 11 e Figura 12, com a primeira apresentando uma visão geral e comparação entre a frota de veículos por estado enquanto a segunda informa mais sobre a frota de veículos no estado do Pará, mas perdendo a visão geral da primeira. O IBGE Cidades também apresenta outros temas, como por exemplo Educação, Economia, Território, População e Trabalho; cada um com gráficos e mapas representativos.

Outro site consultado é o GeoCAPES¹, o qual possui uma interface semelhante com relação à do IBGE Cidades, apresentando um mapa principal e os gráficos de apoio, sendo ativados com um clique em algum elemento do mapa principal. Entretanto, as informações exibidas estão limitadas aos dados acadêmicos obtidos pela CAPES, como por exemplo, a concessão de bolsas de pós graduação, distribuição de docentes e acessos ao portal de periódicos. A Figura 13 mostra a interface do site GeoCAPES.

¹ <https://geocapes.capes.gov.br/geocapes/>

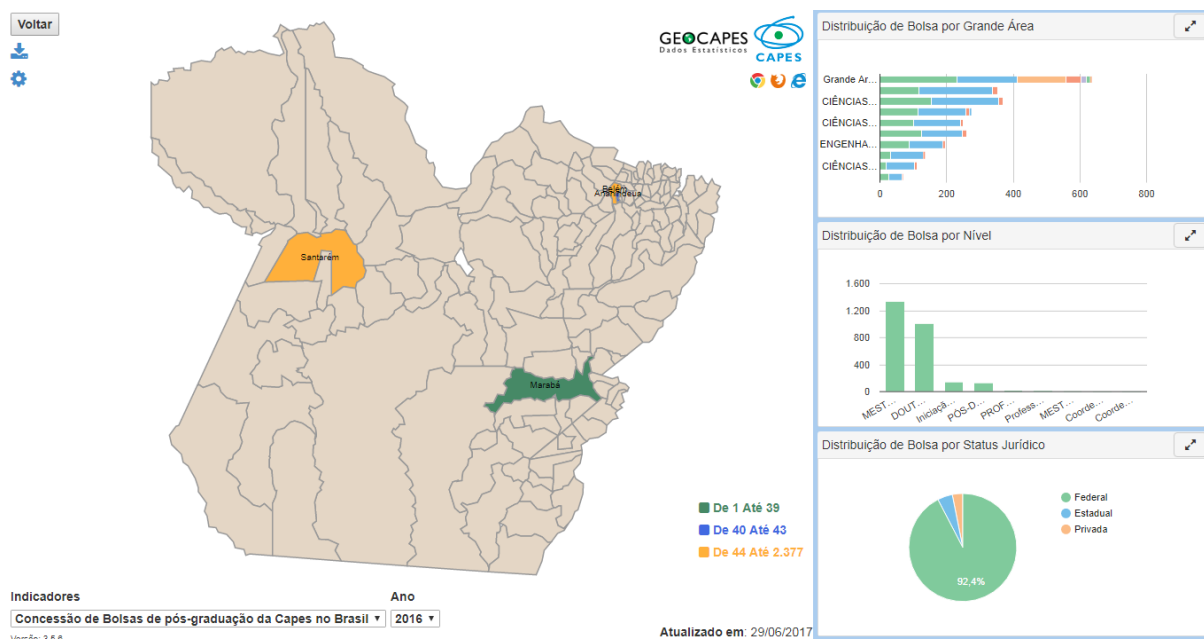
Figura 13 – Página Web do GeoCAPES mostrando a distribuição da Concessão de bolsas de Pós-Graduação no Brasil.



Fonte – Captura de tela do Site GeoCAPES

Além disso, o GeoCAPES possui a interação de exibir mais detalhes do estado selecionado, funcionando como um avanço na hierarquia do mapa para um nível administrativo menor (de país para estado). Por exemplo, ao clicar no estado do Pará na Figura 13 é exibido o respectivo estado com as divisões entre municípios, assim como mostrado na Figura 14.

Figura 14 – Página Web do GeoCAPES mostrando a distribuição da Concessão de bolsas de Pós-Graduação nos municípios do Pará.



Fonte – Captura de tela do Site GeoCAPES

Após a análise das aplicações relacionadas, propõe-se a elaboração de uma arquitetura de uma ferramenta de visualização de mapas *choropleth* contendo as seguintes funcionalidades: exibição de informações de diversos temas (por exemplo: transporte, demografia ou área), possibilidade de configurar as variáveis visuais, além de possuir uma hierarquia entre mapas, um filtro categórico e exibição de detalhes sobre demanda.

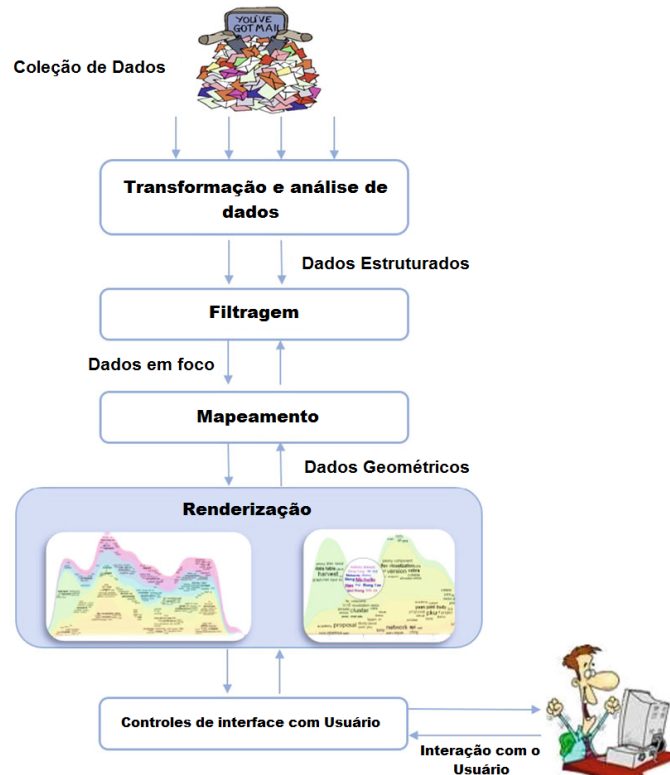
3 Desenvolvimento do Protótipo

Para a implementação das funcionalidades baseadas nas tarefas de visualização definidas em 2.1 e elaboração do protótipo da ferramenta algumas tecnologias *web* foram utilizadas, bem como certas bibliotecas do *Javascript*, sendo descritas nesse capítulo. Além disso a arquitetura e interação entre os componentes do sistema são definidos e posteriormente é exibido um caso de uso demonstrando o funcionamento da ferramenta.

3.1 Arquitetura

A arquitetura utilizada para a elaboração da ferramenta é baseada no *pipeline* (fluxo de atividades) proposto por (LIU et al., 2014), e mostrado na Figura 15. Segundo (LIU et al., 2014), uma aplicação de visualização possui cinco módulos, sendo estes: Transformação e análise dos dados, Filtragem, Mapeamento, Renderização e Controles de interface com o usuário. Os dados, estruturados ou não, servirão de entrada para esse fluxo.

Figura 15 – *Pipeline* de visualização.



Fonte – Imagem adaptada de LIU et al. - *A survey on information visualization: recent advances and challenges*

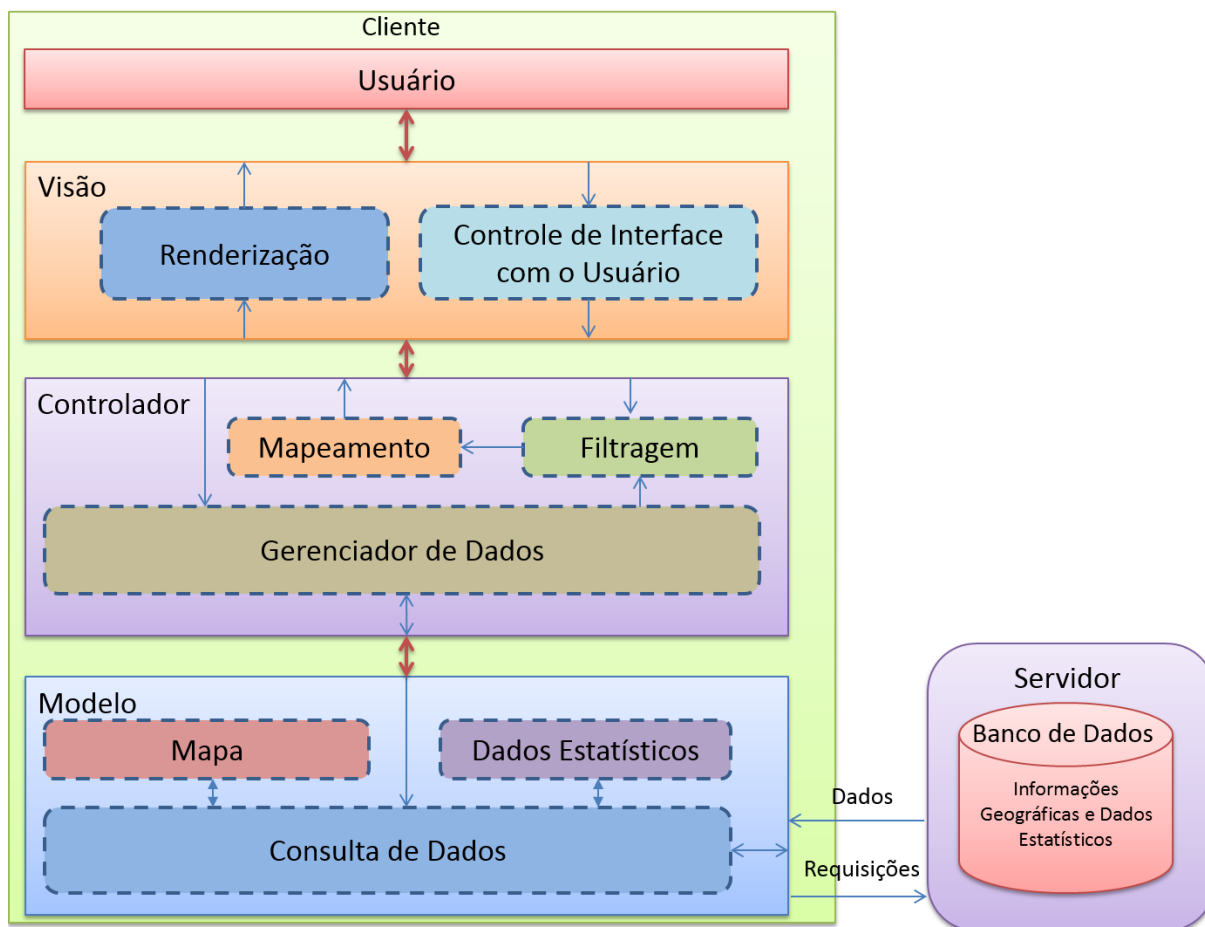
O módulo de Transformação e análise extrai os dados de entrada e converte para

um formato estruturado em que a ferramenta de visualização possa interpretar. Adicionalmente, pode remover ruídos, interpolar valores ausentes e corrigir medições incorretas.

O módulo de Filtragem recebe os dados do módulo de Transformação e análise e seleciona quais dados serão exibidos, passando esses para o módulo de Mapeamento, o qual relaciona os dados em foco (filtrados) em primitivas geométricas e seus atributos, sendo posteriormente transformados em imagens através do módulo de Renderização. Por fim o módulo de Controle de interface com o usuário permite que o mesmo interaja com as imagens geradas, explorando os dados em várias perspectivas (LIU et al., 2014).

Para modularizar a aplicação foram utilizados os modelos Cliente-Servidor, na qual o Cliente efetua um requisição para um Servidor de dados, obtém-os e realiza as tarefas do *pipeline* de visualização; e o modelo MVC (*Model-View-Controller* - Modelo-Visão-Controlador), onde segundo (DEACON, 2005), divide a aplicação em três camadas: o Modelo, representado pelos dados (os mapas e as informações estatísticas), a Visão onde será exibida a interface gráfica da ferramenta, e por último o Controlador sendo responsável por manipular a Visão de acordo com os objetos do Modelo. A Figura 16 representa uma proposta de arquitetura utilizando os modelos supracitados.

Figura 16 – Representação da arquitetura da aplicação, camadas pertencentes e seus componentes.



Fonte – O autor

O pipeline de visualização da Figura 16 começa quando o módulo de Consulta de Dados (responsável por consultar e capturar os arquivos do servidor) obtém os arquivos no servidor de dados, o qual contém os dados demográficos e geográficos. Este módulo ainda converte os dados em um formato estruturado (por exemplo, um objeto na notação JavaScript contendo identificador, nome, lista de pontos e valor associado) para a aplicação e esses arquivos convertidos são enviados para o módulo Gerenciador de Dados. Esta operação é comparável ao módulo de Transformação e análise presente na Figura 15.

Em seguida o módulo Gerenciador de Dados, o qual serve de ponte entre o processamento da aplicação (pois os componentes de Mapeamento e Filtragem também estão no Controlador) e o modelo, transforma os dados em listas ou objetos correspondentes a tecnologia utilizada para interpretá-los. Em resumo, o módulo gerencia a consulta de dados e a transformação desses em uma estrutura interpretável pela tecnologia utilizada.

Após a conversão dos dados eles são enviados para o módulo de Filtragem, onde uma outra estrutura de dados se encarrega de armazenar informações sobre o estado do

filtro para cada elemento; e logo em seguida é enviado para mapeamento, tendo início o processo de classificação dos dados utilizando um algoritmo de classificação.

Os dados já classificados são enviados para o componente de Renderização, o qual irá desenhar as formas geométricas e aplicar as respectivas variáveis visuais, escalando a intensidade das mesmas de acordo com as classes definidas anteriormente. Essa imagem é então enviada para a Interface do Usuário.

O Usuário pode observar a imagem gerada pela renderização e interagir com ela através do módulo de Controle de Interface com o usuário, modificando alguns aspectos da visualização atual, como por exemplo o filtro, onde nesse caso a visualização irá passar novamente pelo componente de filtragem para atualizar o estado do filtro; ou modificar o mapa atual, desta vez o controle de interface com o usuário irá enviar um comando para trocar o mapa exibido para o Gerenciador de Dados. Desta vez irá consultar e obter os respectivos arquivos estatísticos e demográficos, transformá-los em uma estrutura de dados compatível com a tecnologia e seguir o restante do fluxo até a Renderização.

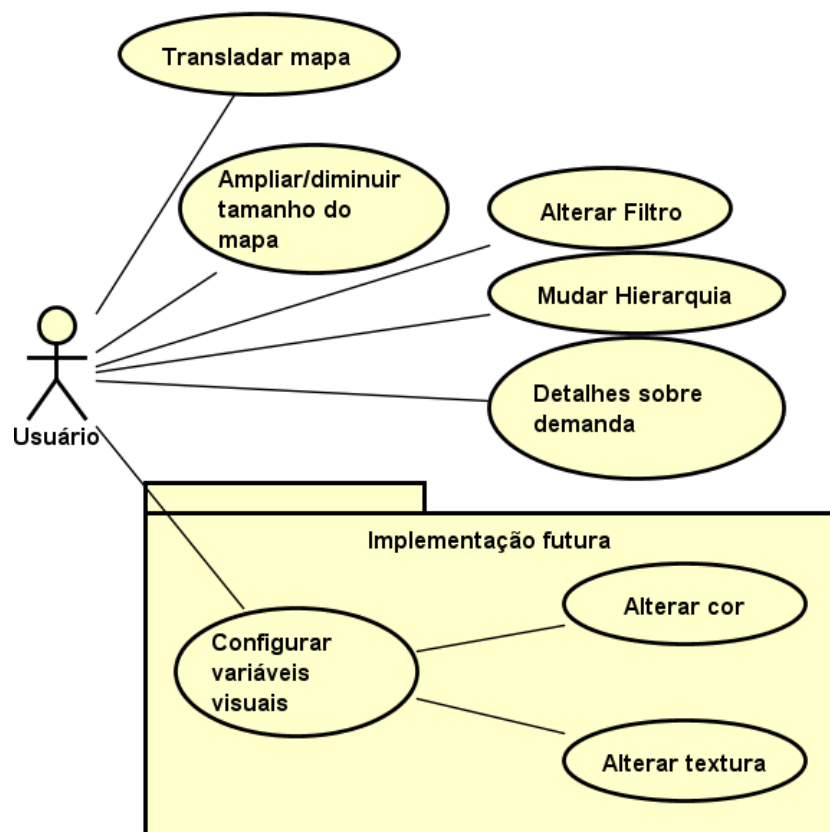
3.2 Modelagem de software

Nesta seção serão exibidos o diagrama de caso de uso mostrando as principais funcionalidades da ferramenta de visualização, o diagrama de classes mostrando as funções dos componentes e suas relações entre si, e, por fim, o diagrama de sequências das principais funcionalidades, detalhando o processo de troca de mensagens entre os componentes.

As possíveis interações que o usuário pode realizar na ferramenta estão descritas no diagrama de Casos de Uso da Figura 17. Nela estão inclusas as funcionalidades fornecidas pela aplicação.

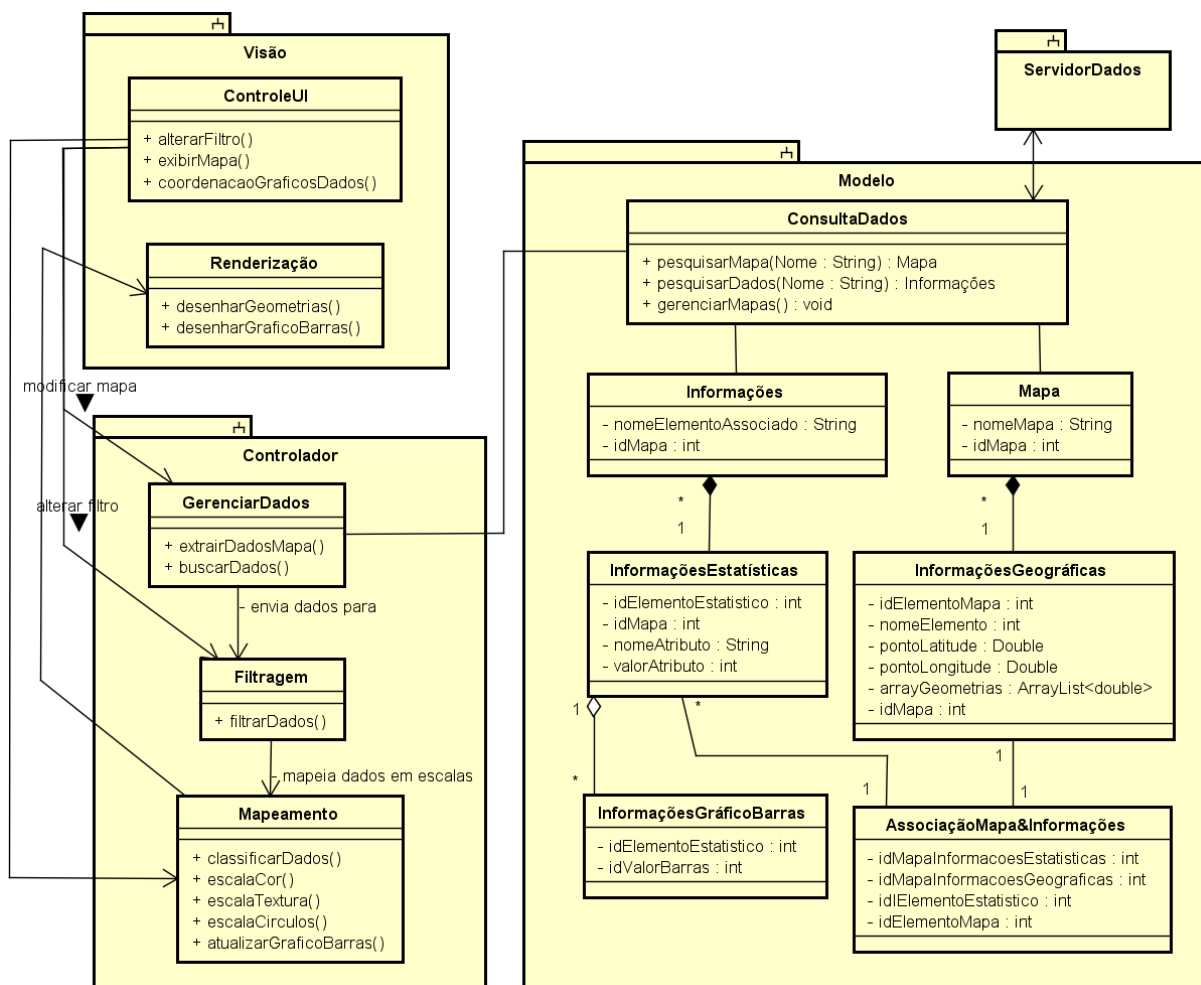
Na Figura 18 há uma representação Orientada a Objetos dos componentes do sistema, descrevendo as classes mais importantes, as associações e sua navegabilidade entre os objetos, os principais métodos e os atributos na camada Modelo.

Figura 17 – Diagrama de Casos de Uso da Ferramenta de Visualização.



Fonte – O autor

Figura 18 – Diagrama de Classes da Ferramenta de Visualização.



Fonte – O autor

Cada mapa possui elementos (estados/municípios/bairros, dependendo da hierarquia) e para cada um há um valor associado nos arquivos estatísticos, sendo relacionados através de outro número de identificação. Um detalhamento do diagrama de classes da Figura 18 é realizado a seguir.

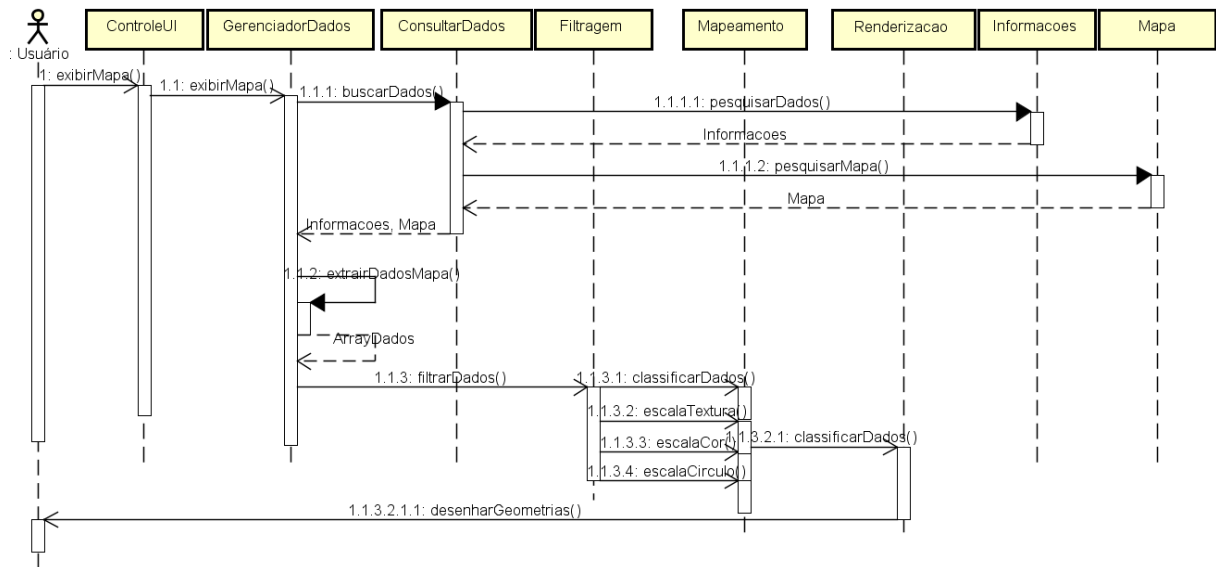
- **Informações Estatísticas** - Compreende os dados sobre um assunto (demografia, área, etc.) que serão exibidos em um elemento no mapa. Possuem um número identificador para serem associados com os elementos dos mapas, o nome do atributo no qual é o dado em questão e seu respectivo valor.
- **Informações Geográficas** - São os dados das geometrias de um elemento de um mapa. Possui um número identificador, um ponto representando o centro do polígono e uma lista de pontos que delimitam a forma geométrica.
- **Associação Mapa e Informações** - Esta classe relaciona quais informações estatísticas pertencem a que elementos geográficos. A relação é feita entre os números

identificadores das informações geográficas e estatísticas, se estiverem iguais então as informações estatísticas são relacionadas ao elemento geográfico.

- Informações do Gráfico de Barras - São detalhes sobre as informações de um elemento estatístico. Por exemplo, informar dados além dos que são exibidos no mapa para um município ou mostrar a evolução dos dados exibidos em um período de tempo.
- Informações - É o conjunto de informações estatísticas a respeito sobre um assunto e relacionadas à um mapa.
- Mapa - É o conjunto de elementos geográficos que formam um mapa.
- Consulta de Dados - É o componente que realiza a consulta dos dados no servidor e transforma em objetos do modelo para serem manipulados pelo Gerenciador de Dados.
- Servidor de Dados - Local onde os arquivos estatísticos e geográficos estão armazenados. São recuperados através da classe Consulta de Dados.
- Gerenciador de Dados - É o módulo que administra a busca dos dados e conversão desses para um formato estruturado que possa ser interpretado pela aplicação.
- Filtragem - É responsável por gerenciar a lista com o estado dos filtros dos elementos do mapa.
- Mapeamento - Realiza as operações de classificação dos dados em escalas e de mapear os dados do gráfico de barras.
- Controle de Interface com o Usuário - Recebe as interações que o usuário executa e modifica a visualização de acordo, como por exemplo alterando o filtro, mudando de mapas ou exibindo informações nos gráficos de barras quando um elemento do mapa é selecionado.
- Renderização - Este módulo desenha o mapa (de acordo com os pontos da geometria), as variáveis visuais (correspondente ao mapeamento dos dados) e o gráfico de barras na tela.

Uma das principais funcionalidades da aplicação é a Exibição de Mapas, sendo assim um detalhamento de como ocorre a troca das mensagens entre os componentes até chegar ao usuário no formato de Imagem é necessário. O Diagrama de Sequência da Figura 19 mostra como esse processo ocorre.

Figura 19 – Diagrama de Sequência da funcionalidade de Exibição de Mapas.

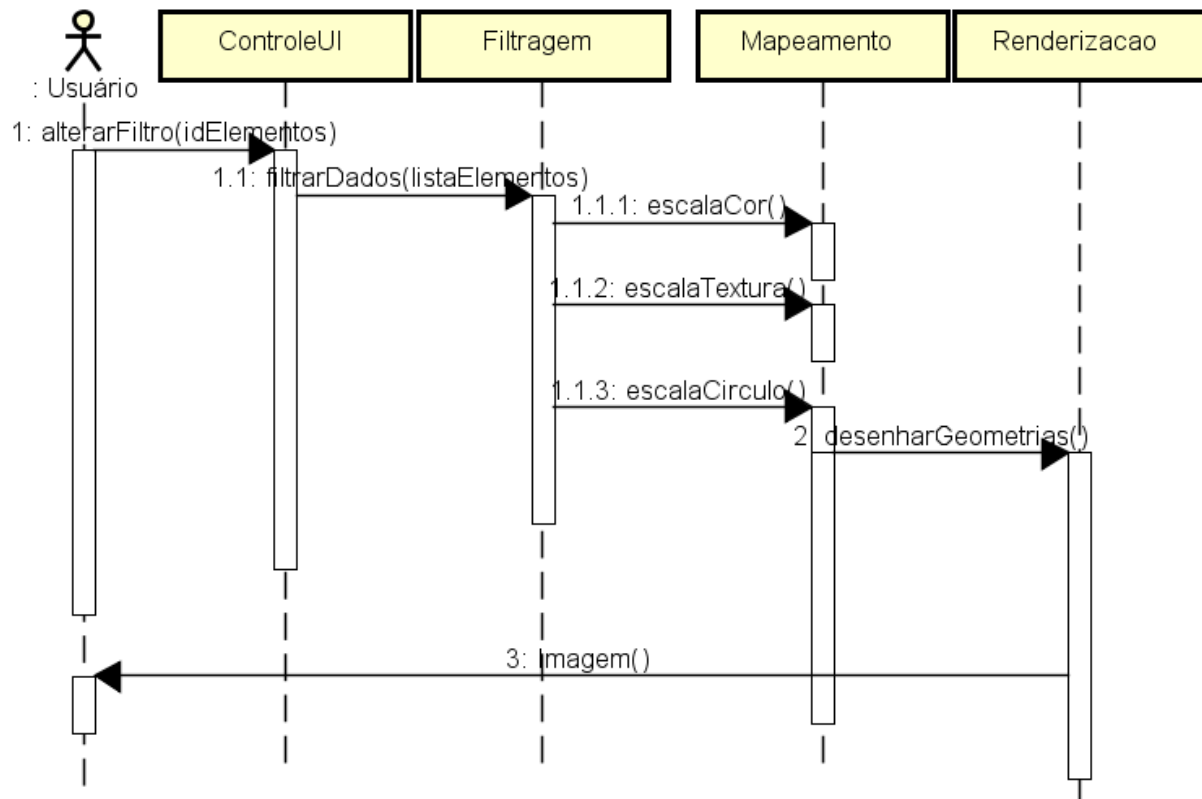


Fonte – O autor

Primeiramente o usuário interage com a aplicação, podendo ser através da inicialização do sistema ou pela mudança de hierarquia. O Controle de Interface do usuário (ControleUI) captura o evento de Exibição de Mapa e repassa para o componente Gerenciador de Dados, com a finalidade de buscar o Mapa e as suas informações estatísticas. O componente de Consulta de Dados realiza uma pesquisa no servidor, captura os dados para o modelo e retorna para o Gerenciador, convertendo-os em um formato estruturado. Depois os dados são filtrados, classificados e mapeados em formas geométricas, no caso do mapa as listas de geometrias são transformadas em linhas, arcos, etc., já para os dados são estabelecidos valores categóricos os quais serão utilizados nas variáveis visuais Cor, Textura e Círculo. Por fim, a Imagem gerada é retornada ao usuário. Esta sequência ocorre também quando o usuário seleciona um elemento do mapa, por exemplo clicando no município Belém, carregando com isso o mapa do respectivo elemento selecionado.

Outra funcionalidade importante é "Alterar Filtro", permitindo que o usuário remova elementos que não são de interesse para o mesmo no momento. Os dados estatísticos estruturados possuem, além dos atributos de escala e número de identificação, um indicador do estado do filtro, informando se o mesmo está ativado ou não. Conforme a Figura 20 mostra, o usuário marca um grupo de elementos contendo o mesmo valor categórico no filtro e o "ControleUI" repassa para a filtragem modificar o estado do filtro de cada, renderizando novamente o mapa (com os valores modificados) e retornando a imagem ao usuário.

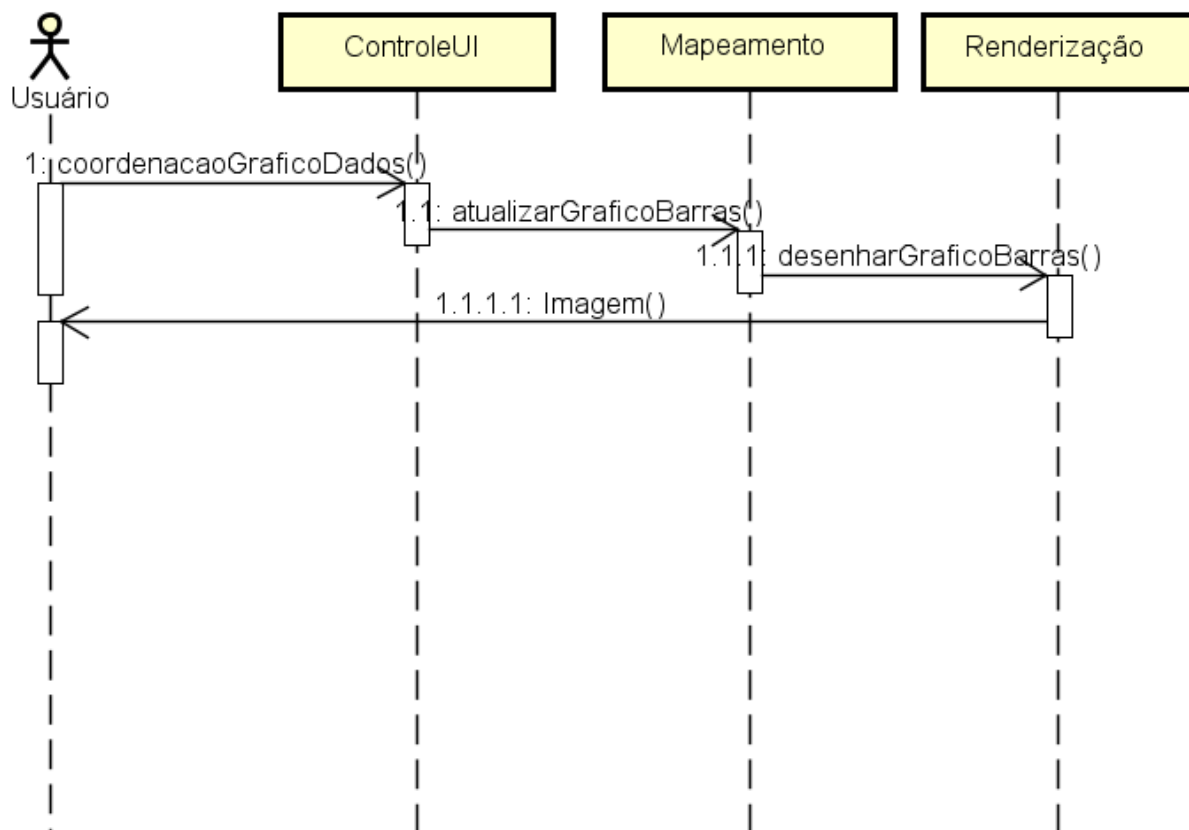
Figura 20 – Diagrama de Sequência da funcionalidade "Alterar Filtro".



Fonte – O autor

Também é possível na aplicação exibir outras informações relacionadas a um elemento do mapa através de um gráfico de barras. Esta é a funcionalidade de "Detalhes sobre demanda" mostrado na Figura 17 do diagrama de casos de uso. As informações do gráfico estão agregadas à classe "Informações Estatísticas" da Figura 18, significando que ao extrair os dados do modelo também estarão inclusas as informações do gráfico de barras, sendo exibidas quando um elemento do mapa é selecionado. A Figura 21 apresenta o usuário ativando o evento de coordenação de dados, fornecendo o identificador do elemento selecionado para o Mapeamento, o qual retorna com todos os elementos do gráfico de barras (quantidade de atributos e os valores associados). Por fim, essas informações são enviadas para Renderização e retornam para o usuário como imagem.

Figura 21 – Diagrama de Sequência da funcionalidade "Detalhes sobre demanda".



Fonte – O autor

3.3 Tecnologias Utilizadas

Neste tópico serão apresentadas as tecnologias escolhidas para fazer parte da implementação do protótipo da ferramenta.

- **HTML** - *HyperText Markup Language* - Utilizada para estruturar as páginas da *Web* para serem interpretadas pelos navegadores (*browsers*). Seu *design* permite que a linguagem evolua com o tempo, permitindo adição de novos recursos e funcionalidades, onde atualmente está na versão 5.1 (W3C, 2016). Nesta ferramenta, o *HTML* possui a função de ser o recipiente para o *JavaScript* no corpo do documento.
- **JavaScript** - É uma linguagem orientada a objetos mais conhecida por ser utilizada na elaboração de *scripts* nas páginas *HTML*, sendo responsável por controlar o comportamento das mesmas e tendo sua execução realizada no lado do cliente, ou seja no navegador do usuário (NETWORK, 2017). Possui um papel importante na ferramenta, pois executa os códigos presentes no controlador.
- **JSON** - *JavaScript Object Notation* - O JSON é, segundo (BRAY T., 2014), uma formatação de arquivo de texto utilizada para armazenar dados estruturados. Isso

torna os dados fáceis de ler e modificar (para seres humanos) e interpretar (para máquinas). Serão utilizadas variações do JSON próprias para codificar dados geográficos: (*GeoJSON* e *TopoJSON*).

- ★ *GeoJSON* - Esta formatação codifica estruturas de dados geográficos utilizando JSON, podendo representar pontos, linhas, polígonos e uma coleção dos mesmos (múltiplos polígonos, por exemplo) (BUTLER et al., 2016). A elaboração do arquivo *GeoJSON* foi realizada com o apoio da ferramenta *Geojson.io*¹ e posteriormente simplificada para o formato *TopoJSON*.
- ★ *TopoJSON* - Uma extensão do *GeoJSON* utilizada para codificar dados geográficos, porém utilizando uma topologia compartilhada, ou seja as geometrias são ligadas entre si por arcos. Devido a criação dos arcos ao oposto de geometrias isoladas, eliminação de redundâncias (como por exemplo as linhas compartilhadas entre elementos são representadas apenas uma vez), e outras técnicas eficientes de codificação, há uma redução no tamanho total do arquivo em até 80% em comparação com o original (BOSTOCK; METCALF, 2013).
- *Geojson.io* - É uma ferramenta utilizada para desenhar (criar), modificar e exportar mapas no formato *GeoJSON*. Os elementos que podem ser criados no mapa são pontos, linhas, polígonos e retângulos. Após o desenho, as formas geométricas podem ter informações adicionadas, como nome e número de identificação.
- *Mapshaper* - Aplicação *Web* utilizada para editar arquivos de geometrias, como por exemplo *GeoJSON* e *TopoJSON*, mas principalmente validar e reparar topologias automaticamente. Permite também a simplificação da geometria dos mapas e adição de *Bounding Boxes* para delimitar os elementos, possibilitando o ajuste da escala e *zoom* para exibição do mapa.
- Um editor de planilhas como por exemplo *Google Docs Sheets* ou *Microsoft Office Excel* para inserir as informações estatísticas e associá-las com os números de identificação dos elementos nos mapas; e posteriormente exportar para um arquivo com valores separados por tabulação (*Tab-Separated Values* ou *.tsv*). Esses arquivos serão interpretados pela ferramenta e armazenados em um objeto estruturado semelhante a um *Array* (um arranjo de elementos) ou lista, com um número de identificação e o valor associado.
- *Scalable Vector Graphics* - Gráficos Vetoriais Escaláveis ou SVG são representações utilizando XML (Linguagem de Marcação Extensível) de imagens, gráficos ou até mesmo animações. Estes arquivos contém informações das formas geométricas que

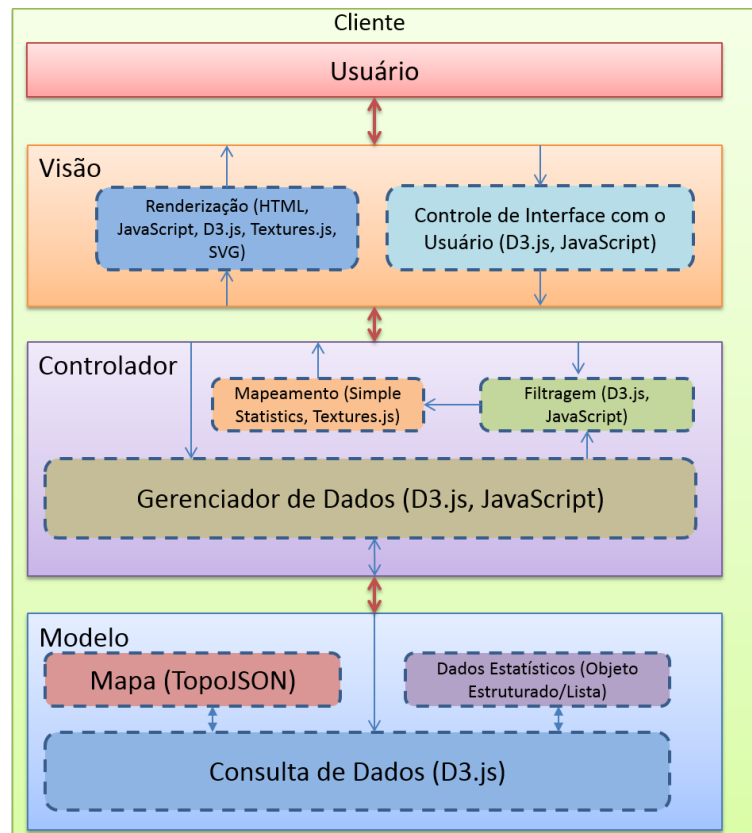
¹ <http://geojson.io/>

compõem as figuras (tamanho e cor, por exemplo). Podem ser escaláveis para qualquer tamanho do elemento ou tela, devido ao elemento ser redesenhado em tempo de execução (PENG, 2000).

- *D3.js* ou - *Data-Driven Documents* - A biblioteca responsável por mapear os dados estatísticos e geoespaciais, manipular a estrutura dos elementos da página *Web* e gerar o SVG dos gráficos na página *HTML* (BOSTOCK; OGIEVETSKY; HEER, 2011).
- *Simple Statistics* - Biblioteca utilizada por implementar a função de classificação *Jenks Natural Breaks* utilizada na ferramenta, mas também possui outros métodos estatísticos, como por exemplo Distribuições, Regressão Linear e outros Classificadores (MACWRIGHT, 2016).
- *Textures.js* - Essa biblioteca proporciona a criação de texturas para serem exibidas nos mapas. Podem ser modificadas nas mesmas as formas geométricas que aparecem nas texturas (linhas, círculos ou um padrão personalizado) e suas propriedades (tamanho, direção, cor de fundo, cor da forma) (SCALCO, 2015).

A Figura 22 mostra uma releitura da arquitetura representada na Figura 16, associando os componentes e as tecnologias utilizadas em cada um.

Figura 22 – Arquitetura da ferramenta com as tecnologias utilizadas em cada componente.



Fonte – O autor

3.4 Aspectos de Implementação

Nesta seção será explicado quais dados de teste foram utilizados, como foram obtidos e aplicados na ferramenta.

3.4.1 Obtenção de dados

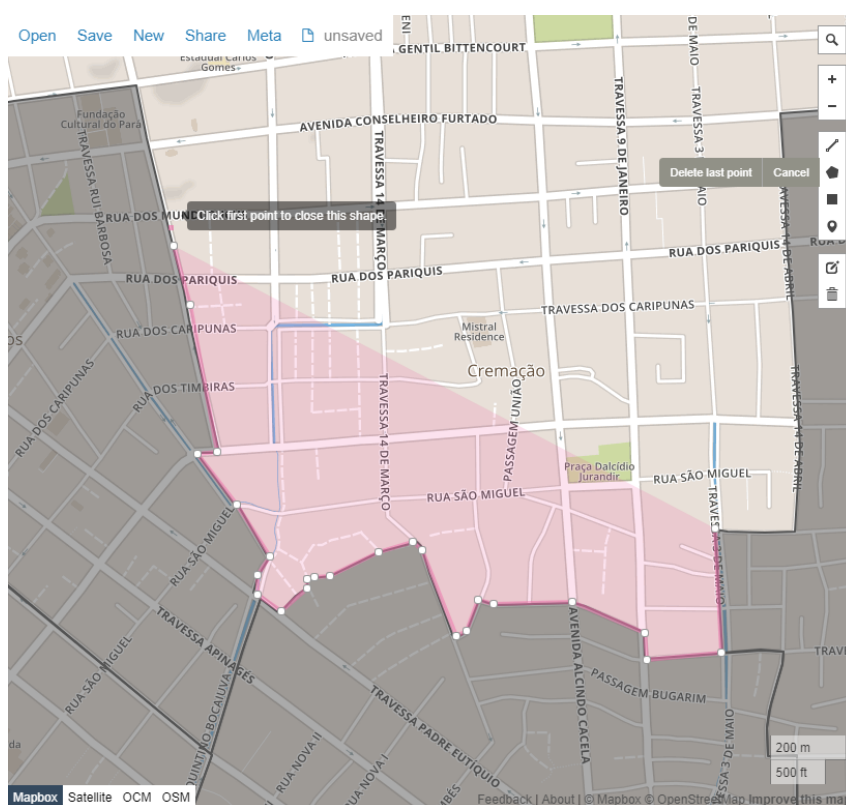
Neste trabalho foram utilizados os mapas do Brasil, municípios do Pará e dos bairros de Belém, junto com as informações de população e área de cada um.

Para a ferramenta de visualização de mapas são necessários os dados geográficos do mapa em questão e os dados com as informações à serem exibidas (por exemplo demografia, área, etc.). Como caso de teste inicial foram utilizados os bairros do município de Belém, o mapa do estado do Pará (com os municípios) e por fim o mapa do Brasil.

A melhor situação seria que estes dados com as geometrias já existissem (como por exemplo os mapas do Brasil e dos municípios do Pará, encontrados no seguinte repositório

na *Internet: Geodata - BR Brasil*¹), mas no caso dos bairros de Belém não foi possível, sendo assim houve a necessidade de criar essas geometrias - podendo ser chamados também de *features* (características) - utilizando a ferramenta *geojson.io*², descrita na subseção 3.3, com um exemplo de utilização na Figura 23.

Figura 23 – Interface do *geojson.io* durante a adição de um polígono representando um bairro de Belém.



Fonte – Site GeoJSON.io

A delimitação dos polígonos no *geojson.io* foi realizada seguindo os mapas dos bairros disponíveis no *site* da Prefeitura de Belém³, mostrado na Figura 24, com o apoio da descrição textual dos limites (lei dos bairros) disponível no *site* Guia do Eleitor⁴, com um exemplo exibido na Figura 25. As geometrias de Belém foram criadas pelo autor e estão disponíveis no repositório *Git Hub* pessoal⁵.

¹ <https://github.com/tbrugz/geodata-br>

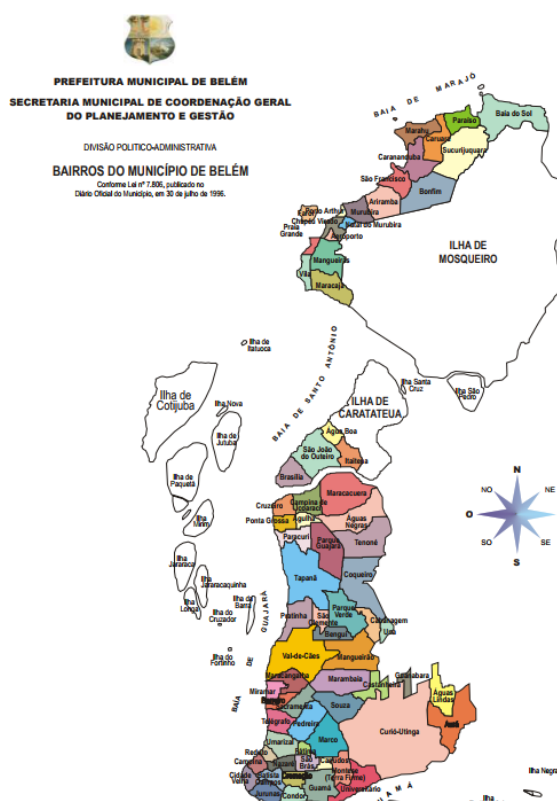
² <http://geojson.io>

³ <http://www.belem.pa.gov.br/app/c2ms/v/?id=18&conteudo=4758>

⁴ <http://www.guiadoeleitor.com.br/bairros/bairros.htm>

⁵ <https://github.com/marcosbenaion/Choropleth-Maps>

Figura 24 – Bairros do município de Belém.



Fonte – Prefeitura Municipal de Belém

Figura 25 – Trecho da página Guia do Eleitor descrevendo os limites do bairro Cidade Velha.

Bairros de Belém

Guia do > Guia do **Cidade Velha** > Guia do

Distrito Administrativo
DABEL - Belém

Limites (Lei dos Bairros): Compreende a área envolvida pela poligonal que tem início na interseção da margem oriental da Baía do Guajará com a Av. Portugal, segue por esta até a Rua João Diogo, a partir da qual passa a se chamar Rua Desembargador Inácio Guilhon, segue por esta até a Av. Almirante Tamandaré, onde passa a receber a denominação de Av. 16 de Novembro, segue por esta até sua interseção com a Rua Cesário Aivim, flete à direita e segue por esta e por seu prolongamento até encontrar a margem direita do Rio Guamá, Flete à direita e segue por este até o início da poligonal.

Fonte – Guia do Eleitor

Já os dados estatísticos dos estados do Brasil e dos municípios do Pará contendo população total, área, entre outros, foram obtidos através do IBGE - Instituto Brasileiro de Geografia e Estatística¹. O IBGE fornece um serviço *Web* o qual permite consultar esses dados coletados do último censo demográfico e estimativas para os anos em que o censo não foi realizado. Com relação as informações dos bairros de Belém foi utilizado

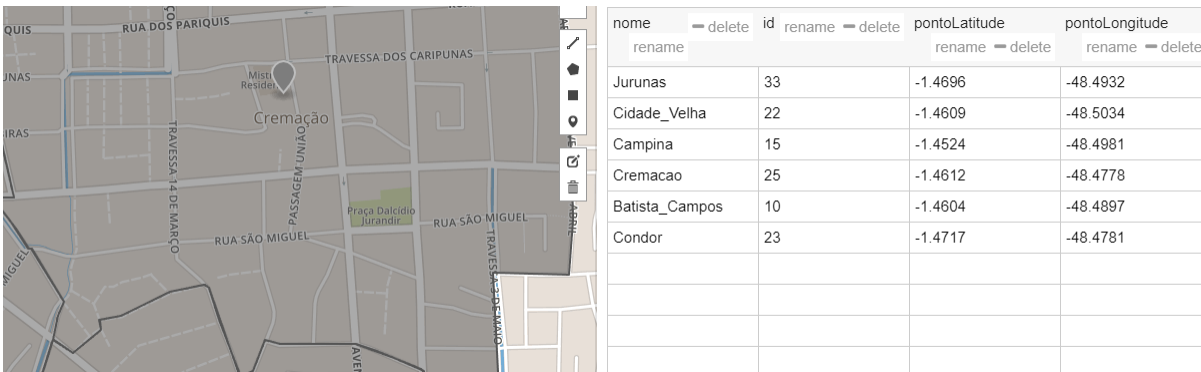
¹ <https://www.ibge.gov.br/>

o documento "Anuário Estatístico do Município de Belém 2011"², o mais atualizado ao momento.

3.4.2 Tratamento e análise dos dados

De posse das geometrias dos Estados do Brasil/Municípios do Pará/Bairros de Belém e as respectivas informações sobre demografia, área, etc, inicia a etapa de formação dos dados. Utilizando as geometrias e a ferramenta *geojson.io* é definido para cada elemento do mapa (Estado, Município ou Bairro) o seu respectivo nome, sem caracteres especiais e espaços substituídos pelo caractere "_" *underscore*, um número de identificação ou ID e um ponto representando as coordenadas de Latitude e Longitude, assim como mostrado na Figura 26. Ao final da edição deverá ser feita a exportação do mapa para um arquivo no formato *GeoJSON* com o nome "*collection.json*" para padronização do nome do objeto mapa acessado pelo *Javascript*.

Figura 26 – Definindo valores para os campos nome, id (número de identificação), ponto latitude e longitude para cada bairro de Belém.



nome	id	pontoLatitude	pontoLongitude
Jurunas	33	-1.4696	-48.4932
Cidade_Velha	22	-1.4609	-48.5034
Campina	15	-1.4524	-48.4981
Cremacao	25	-1.4612	-48.4778
Batista_Campos	10	-1.4604	-48.4897
Condor	23	-1.4717	-48.4781

Fonte – Site GeoJSON.io

Ainda no contexto da geometria dos elementos deve-se realizar o processo de validação da topologia, isto é verificar se não há erros de topologia, como por exemplo polígonos desconexos ou com múltiplos polígonos em um único *feature*. Utilizando a ferramenta *geojson.io*, não há problemas de topologia ao criar um mapa, mas é necessário ter atenção ao utilizar arquivos de geometria já existentes. Com isso, o aplicativo *Web Mapshaper* foi utilizado, pois fornece a detecção e correção automática destes erros, além de adicionar as delimitações dos elementos (*Bounding Boxes*) para proporcionar um *zoom* automático a todos os *features*.

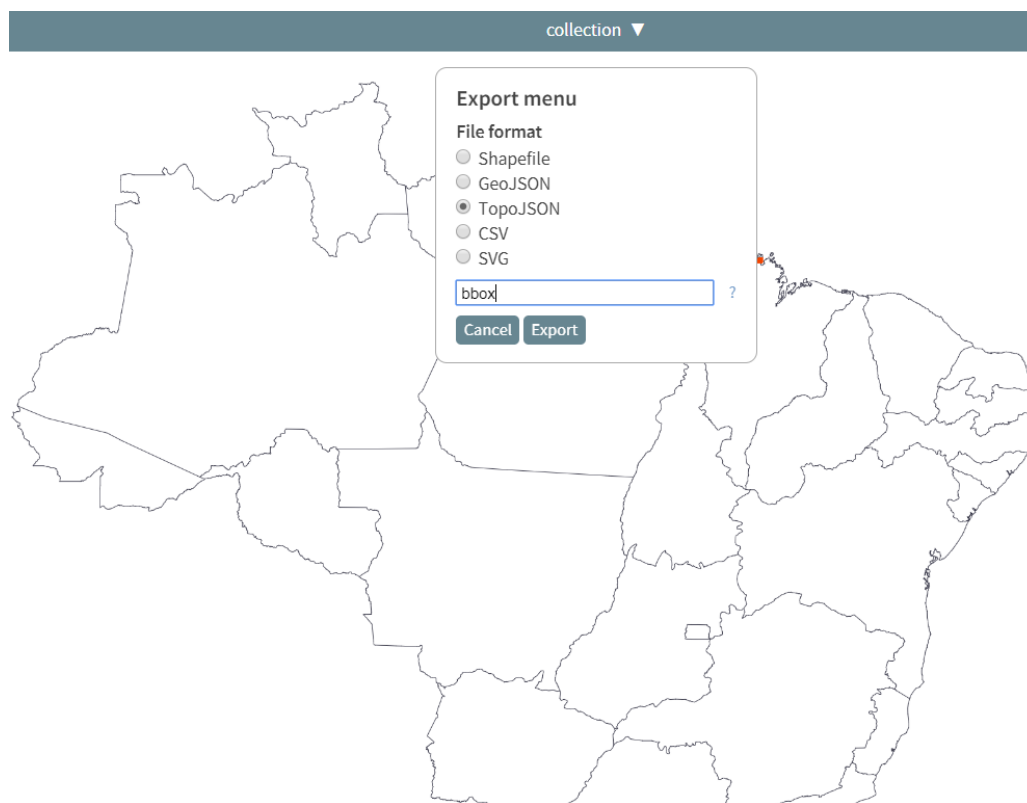
O procedimento realizado inclui o envio (*upload*) do mapa no *Mapshaper*¹ e a exportação do mapa para um arquivo no formato *TopoJSON*, com opção de linha de comando "*bbox*". A Figura 27 mostra esse último passo antes de clicar no botão "*Export*".

² http://www.belem.pa.gov.br/app/ANUARIO_2011/ANUARIO%202011%20COMPLETO.pdf

¹ <http://mapshaper.org/>

Por fim, o arquivo pode ser renomeado para o seu nome definitivo (nome do país, estado ou bairro; seguindo do nome do formato ".json"). É necessário observar os nomes na hierarquia dos mapas, pois a ferramenta irá buscar o arquivo de acordo com o texto definido no campo NOME do respectivo elemento, por exemplo, se no mapa dos estados do Pará um município possui o NOME "Sao_Felix_do_Xingu", o sistema irá buscar pelo arquivo "Sao_Felix_do_Xingu.json".

Figura 27 – Mapa do Brasil pronto para ser exportado em um arquivo *TopoJSON*.



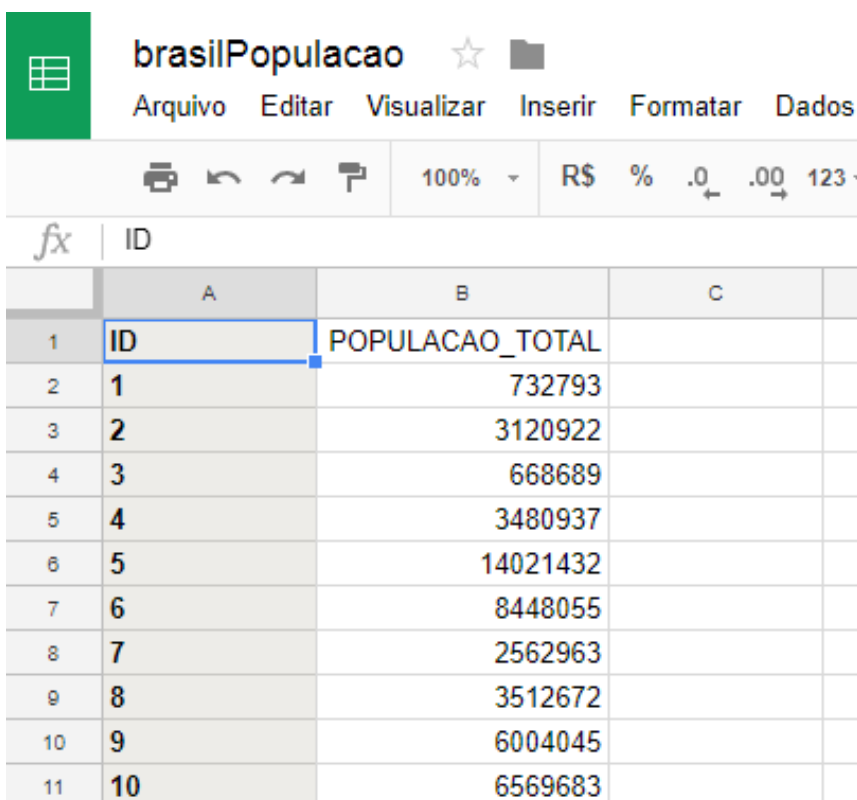
Fonte – Site Mapshaper

Para os dados estatísticos, o formato utilizado pela ferramenta é *.tsv - Tab-separated values*, ou Valores Separados por Tabulação - o qual pode ser obtido utilizando um programa editor de planilhas como o *Microsoft Office Excel*, *Google Docs* ou até mesmo em um editor de texto. Os dados devem ser inseridos utilizando duas colunas, a primeira nomeada ID e a segunda utilizando o tipo de dado que será utilizado, por exemplo POPULACAO_TOTAL ou AREA. Cada ID neste documento corresponde ao mesmo valor ID em um elemento no arquivo com as geometrias do mapa.

Por exemplo, se o estado Goiás estiver com o ID igual a 9 no arquivo de geometrias dos estados do Brasil, para adicionar o valor de população total de Goiás no arquivo de planilha o mesmo ID de número 9 deve ser utilizado, seguido da população na coluna POPULACAO_TOTAL. Ressaltando que cada atributo deve estar em um arquivo separado com o nome da hierarquia a qual o elemento pertence seguido da palavra "Populacao" ou

"Area", com a primeira letra em maiúsculo, por exemplo: "BrasilArea.tsv". A Figura 28 mostra um exemplo de uma planilha com a população dos estados do Brasil utilizando o *Google Docs*.

Figura 28 – Planilha contendo população e 'id' dos estados brasileiros.



The image shows a screenshot of a Google Docs spreadsheet titled "brasilPopulacao". The spreadsheet has a menu bar with "Arquivo", "Editar", "Visualizar", "Inserir", "Formatar", and "Dados". Below the menu bar is a toolbar with icons for print, undo, redo, and a zoom level of 100%. The spreadsheet itself has a header row with columns labeled "ID" and "POPULACAO_TOTAL". The data rows show the following values:

	A	B	C
1	ID	POPULACAO_TOTAL	
2	1	732793	
3	2	3120922	
4	3	668689	
5	4	3480937	
6	5	14021432	
7	6	8448055	
8	7	2562963	
9	8	3512672	
10	9	6004045	
11	10	6569683	

Fonte – Site *Google Docs* Planilhas

Caso as informações demográficas necessitem ser atualizadas, o usuário pode editar a planilha contendo os dados. Um mapa mais os dados estatísticos é definido como pronto ou concluído quando possui os arquivos nos formatos ".json" e ".tsv", tais como definidos anteriormente.

3.4.3 Limitações das pesquisas

Devido a natureza da pesquisa depender dos dados relacionados aos mapas, tanto estatísticos quanto as geometrias, alguns elementos podem necessitar de uma pesquisa em documentos de órgãos públicos, por exemplo, a Prefeitura de Belém para encontrar o mapa dos bairros de Belém, e, principalmente, da existência desses. Outro ponto é com relação as datas das informações estatísticas, as quais podem estar desatualizadas e não refletir o cenário atual no mapa (por exemplo, censo populacional em 2011 ser o mais atualizado), tendo que utilizar projeções ou estimativas.

3.5 Funcionalidades

Nessa seção as funcionalidades da ferramenta serão detalhadas em conjunto com as tecnologias utilizadas, explicando como usá-los e seus comportamentos. Algumas podem ser acessadas durante a utilização da aplicação, outras necessitam de alterações nos arquivos JavaScript e/ou manipulação dos dados obtidos (*GeoJSON* ou arquivo de tabulação no formato *.tsv*). As funcionalidades são listadas a seguir.

- Ampliar/Diminuir tamanho do mapa - Também conhecido como "Zoom" ou "Zooming", corresponde à ação de aproximar ou afastar os elementos do mapa para o campo de visão do usuário. A biblioteca *D3.js* possui dois métodos que manipulam esta interação: *d3.zoom* e *d3.event.transform*. Pode ser utilizado através da movimentação da roda do mouse para frente ou para trás (*scroll*) ou duplo-clique em um espaço vazio no mapa.
- Transladar Mapa - Também chamado de *Pan*, é o evento de clicar e arrastar o mapa na visualização, deslocando seus elementos no espaço do SVG em relação ao movimento do mouse. Pode ser realizado em qualquer direção.
- Alterar Filtro - Consiste em remover um grupo de elementos da visualização presente. Os elementos do grupo são definidos através de um algoritmo de classificação, como por exemplo o *Jenks Natural Breaks*. Ao clicar em um dos botões no lado esquerdo do mapa os elementos serão removidos (os respectivos valores do *Array* de filtros são configurados para zero) e remapeados e renderizados novamente. Como o filtro foi desativado para os elementos selecionados a renderização os mostra na cor cinza. Clicando novamente no botão irá reativar os elementos no mapa. Atualmente está disponível apenas para filtrar uma categoria de dados.
- Mudar Hierarquia - Acessa o mapa (se houver) de um elemento selecionado do mapa anterior. Essencialmente chama "Exibir Mapa" para o elemento clicado.
- Detalhes sobre demanda - É um evento que exhibe no gráfico de barras à direita (na aplicação) as informações disponíveis sobre o elemento selecionado no mapa. O evento é capturado pelo método *onclick* do arquivo *JavaScript* responsável pelo mapa e chama o método de desenho do gráfico de barras, em seu respectivo arquivo *JavaScript*.
- Configurar variável visual de cor - As cores exibidas nos elementos do mapa da ferramenta podem ser alteradas no arquivo *JavaScript* do mapa, na variável *colors*, a qual é uma lista de cores e texturas associadas a um determinado valor de grupo (esses números são gerados através do algoritmo de classificação *Jenks Natural Breaks*). As cores estão definidas no formato RGB, o qual define a intensidade das cores

Vermelho, verde e azul para criar a cor RGB. A Figura 29 mostra um exemplo de uma lista de cores com tons verdes (do verde claro ao verde escuro).

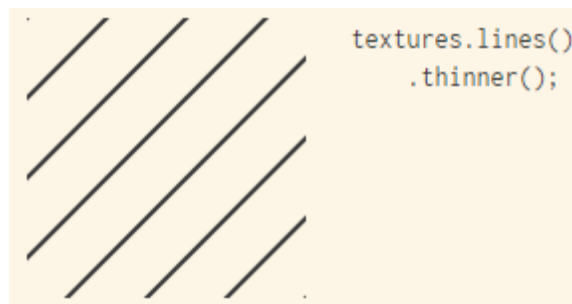
Figura 29 – Lista de cores RGB com tom verde, partindo do verde claro ao verde escuro, respectivamente de baixo para cima.

```
var colors = [{ colorMap: "rgb(247,251,255)" },  
              { colorMap: "rgb(222,235,247)" },  
              { colorMap: "rgb(198,219,239)" },  
              { colorMap: "rgb(158,202,225)" },  
              { colorMap: "rgb(107,174,214)" },  
              { colorMap: "rgb(66,146,198)" },  
              { colorMap: "rgb(33,113,181)" },  
              { colorMap: "rgb(8,81,156)" },  
              { colorMap: "rgb(8,48,107)" }  
            ];
```

Fonte – O autor

- Configurar variável visual de textura - Da mesma forma que na alteração das cores, as texturas estão definidas na variável *texture*. Entretanto, nelas aparecem apenas as instâncias, ou seja, as texturas primeiro precisam ser configuradas em aspectos como por exemplo: qual forma geométrica será apresentada (linhas, círculos, quadrados ou um padrão personalizado), tamanho, orientação e cores; para depois serem associadas em um elemento do mapa. Após a criação da textura é possível substituí-la na variável *texture*. A Figura 30 mostra a criação de uma textura simples com linhas na diagonal.

Figura 30 – Exemplo de criação de uma textura utilizando a biblioteca *Textures.js*.



Fonte – Site *Textures.js*

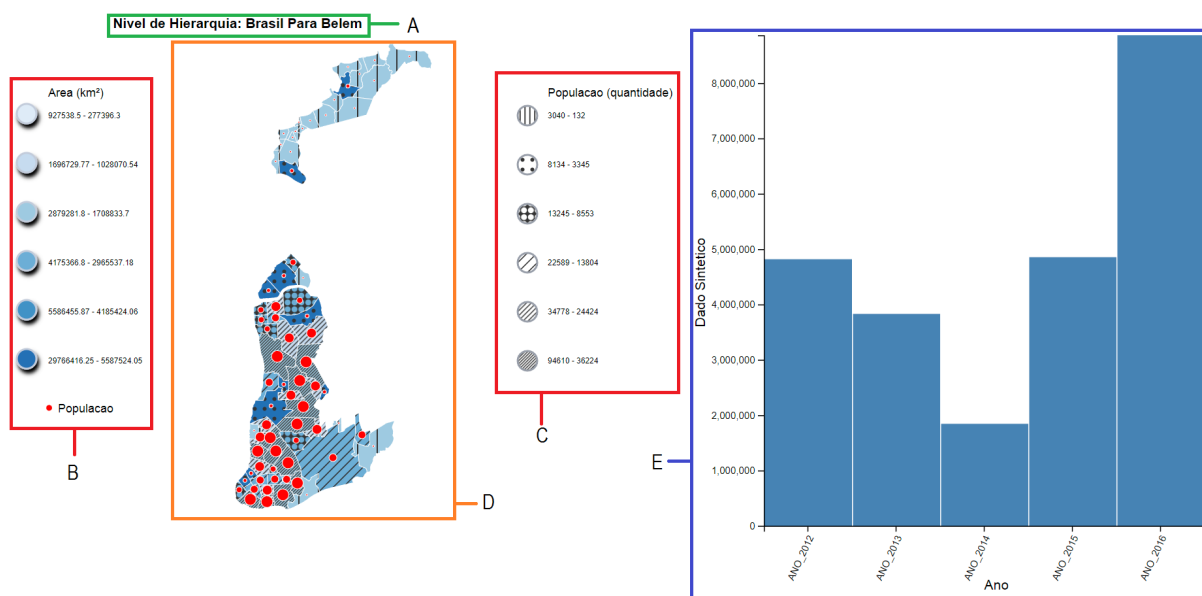
3.6 Protótipo

Nesta seção é apresentada a aplicação através de capturas de tela, exibindo o caso de uso dos mapas do Brasil, Pará e Belém para demonstrar as funcionalidades definidas na seção 3.5.

A implementação da ferramenta contou com a utilização de um editor de texto e um servidor na linguagem *python* hospedado localmente. O servidor irá fornecer o acesso aos arquivos entre o *JavaScript* e o computador através do método GET do protocolo HTTP (*Hypertext Transfer Protocol*).

A interface principal da ferramenta é mostrada na Figura 31. Nela estão marcadas cinco áreas: Na área A, no canto superior esquerdo, aparece a hierarquia atual exibida no mapa, sendo possível navegar por ela para voltar a um mapa desejado; os filtros categóricos e legendas, localizados nas áreas B e C, onde cada área possui um grupo de seis botões, um selecionando as cores (área B) e outro selecionando as texturas (área C), associados aos grupos de dados (organizados pelo algoritmo de classificação); o mapa interativo ao centro na área D e por fim a área E do gráfico de barras à direita. As cores estão associadas às áreas dos elementos no mapa, enquanto as texturas e os círculos estão associados à população.

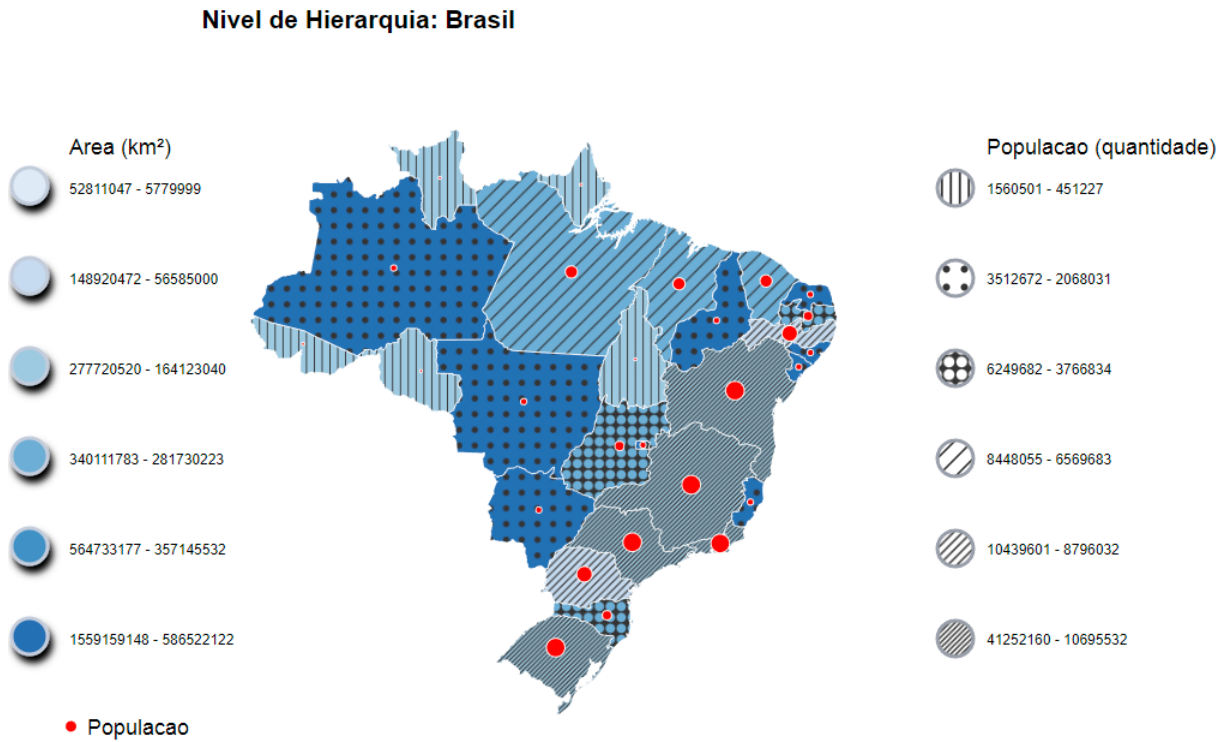
Figura 31 – Visão geral do protótipo.



Fonte – O autor

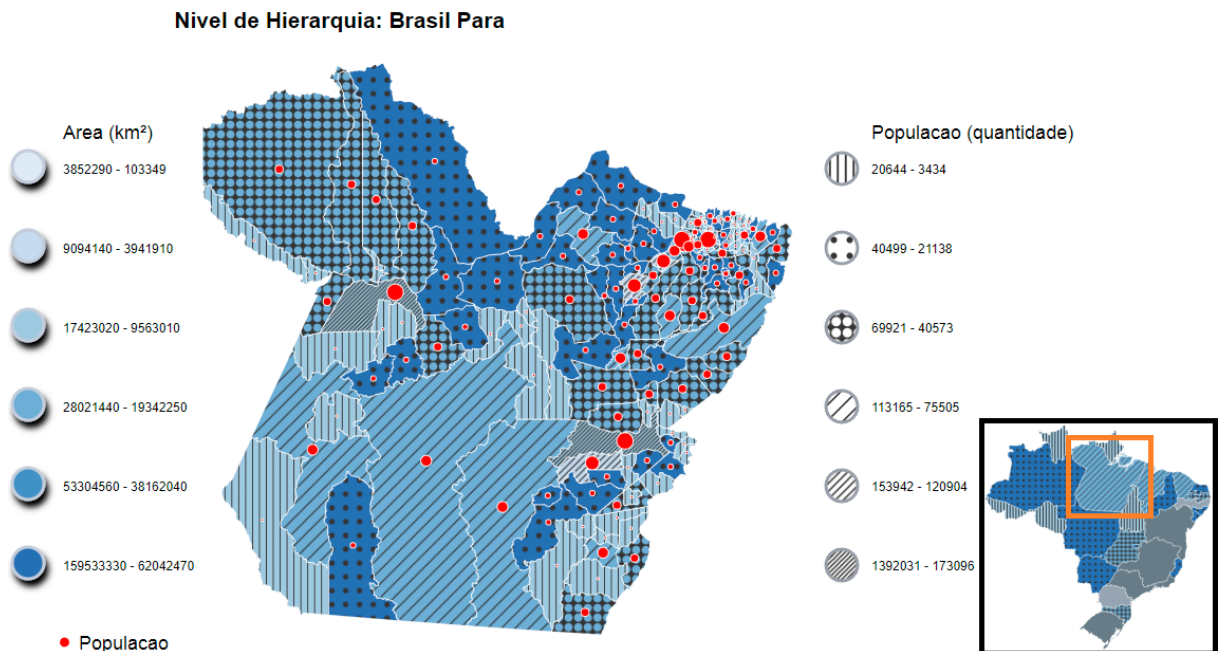
A primeira tarefa é a de exibição do mapa, na qual a aplicação começa no primeiro nível de hierarquia, no mapa do Brasil. A Figura 32 mostra esse evento. Ao clicar no estado do Pará (o qual aparece o texto informativo do número identificador do elemento, o nome "Pará", a população e a área do estado na imagem) a ferramenta irá carregar o respectivo mapa, exibido na Figura 33. Note que ainda é possível voltar ao mapa do Brasil clicando no nome "Brasil" no canto superior esquerdo da tela.

Figura 32 – Mapa do Brasil.



Fonte – O autor

Figura 33 – Mapa do estado do Pará (após o clique no estado do Pará no mapa do Brasil).

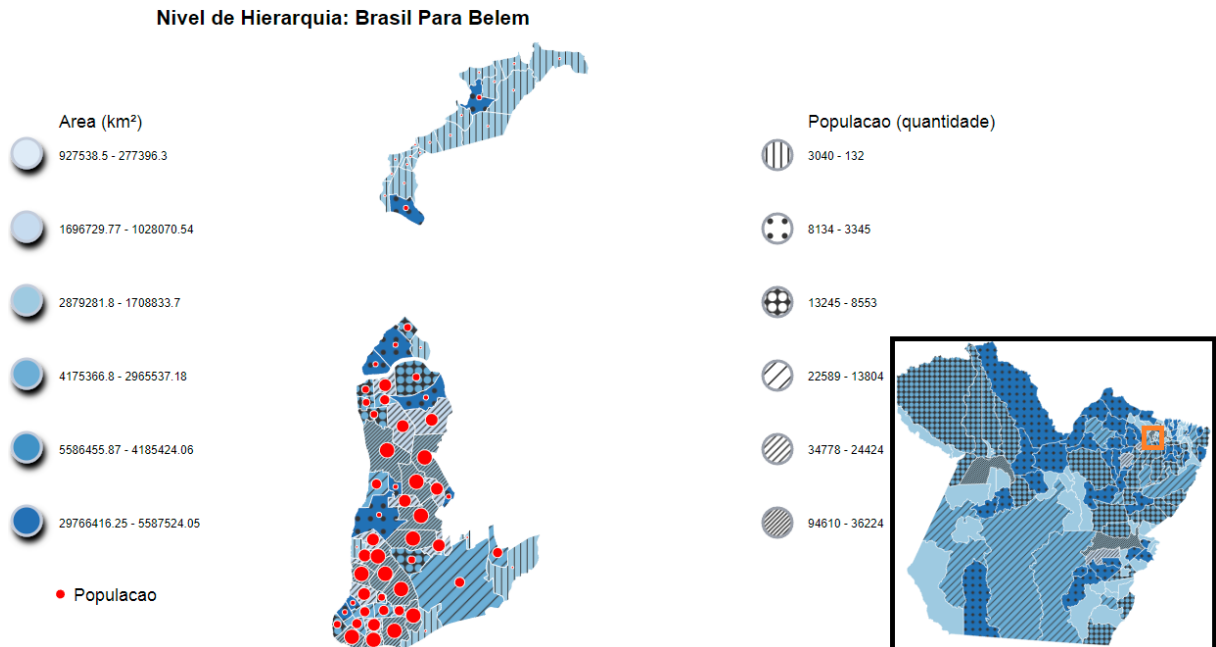


Fonte – O autor

No mapa do estado do Pará ainda é possível prosseguir em mais um nível de hierarquia clicando no município de Belém. Como os pontos vermelhos sobrepõem o tamanho

dos municípios ao redor, um evento para os pontos foi criado para permitir a seleção de um município através do círculo associado. Ao clicar no círculo que representa Belém no mapa, o mapa dos bairros de Belém é exibido, conforme a Figura 34 mostra.

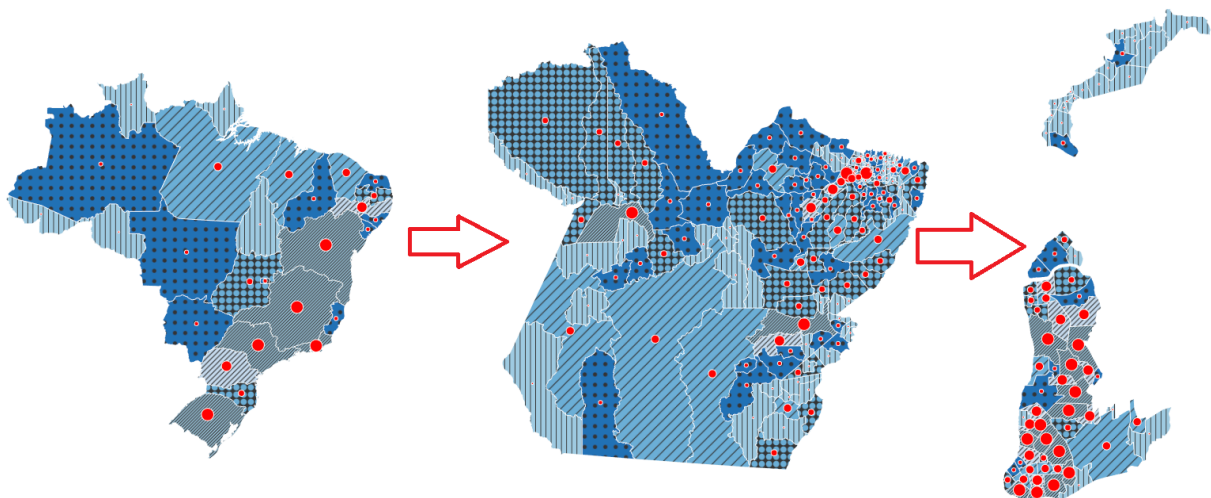
Figura 34 – Mapa dos bairros do município de Belém.



Fonte – O autor

Para recapitular a hierarquia percorrida, a Figura 35 mostra a transição entre os mapas do Brasil, Pará e Belém.

Figura 35 – Hierarquia percorrida até chegar ao mapa dos bairros de Belém.

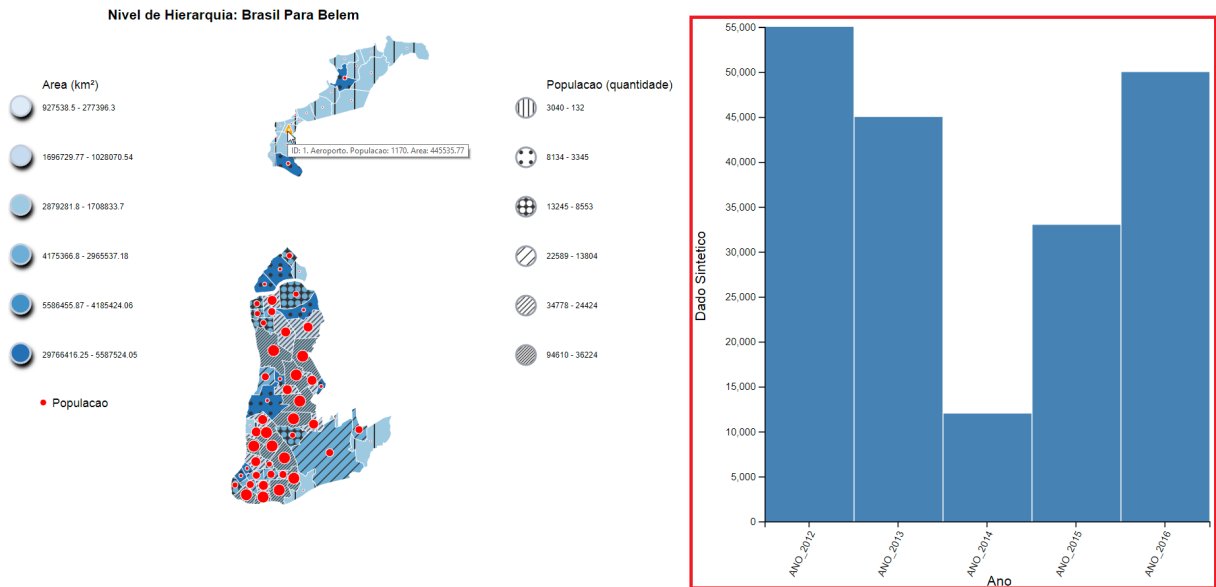


Fonte – O autor

Outro recurso do protótipo é a exibição de dados específicos em um gráfico de barras. Nesse caso de uso a coordenação entre mapa e gráfico funciona apenas no nível

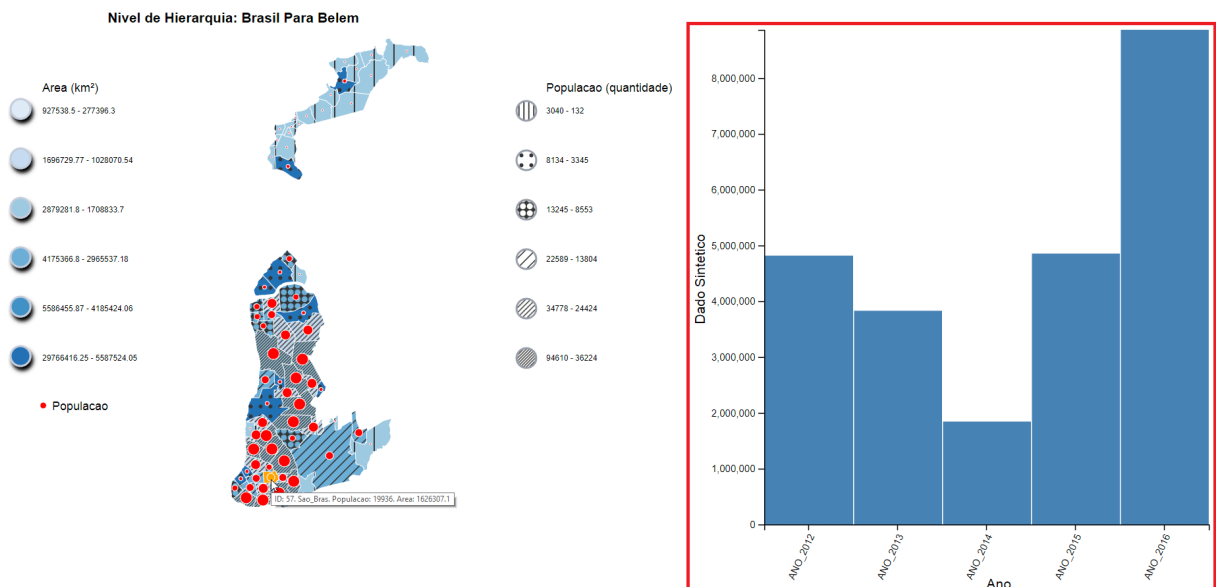
hierárquico do mapa de Belém. As Figuras 36 e 37 mostram o funcionamento desse mecanismo utilizando um dado fictício com nome de "Dado arbitrário" em função dos anos 2012 a 2016. Os bairros selecionados são, respectivamente, "Aeroporto" e "São Brás".

Figura 36 – Dados exibidos sobre o bairro Aeroporto.



Fonte – O autor

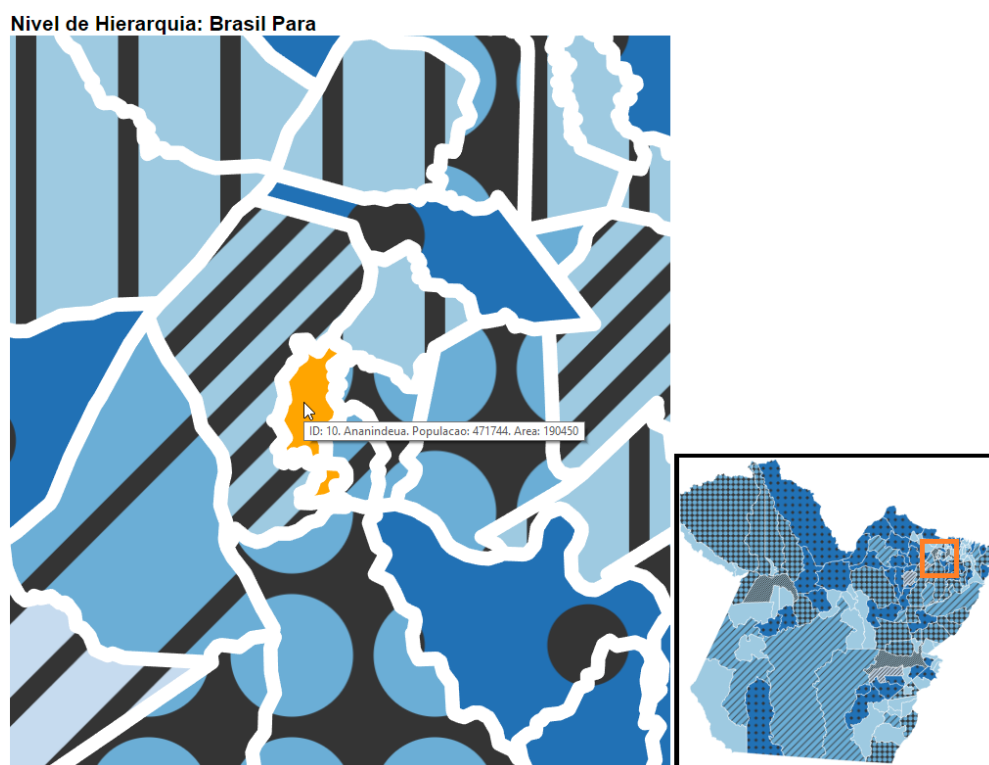
Figura 37 – Dados exibidos sobre o bairro São Brás.



Fonte – O autor

Um dos problemas mencionados anteriormente seria a dificuldade em visualizar um município muito pequeno ou perto de outros. Com isso a funcionalidade de *zoom* permite ampliar a imagem, proporcionando uma melhor visão dos detalhes do mapa. A Figura 38 demonstra a aplicação desta técnica.

Figura 38 – Ampliação do mapa dos municípios do Pará, com destaque para Ananindeua.

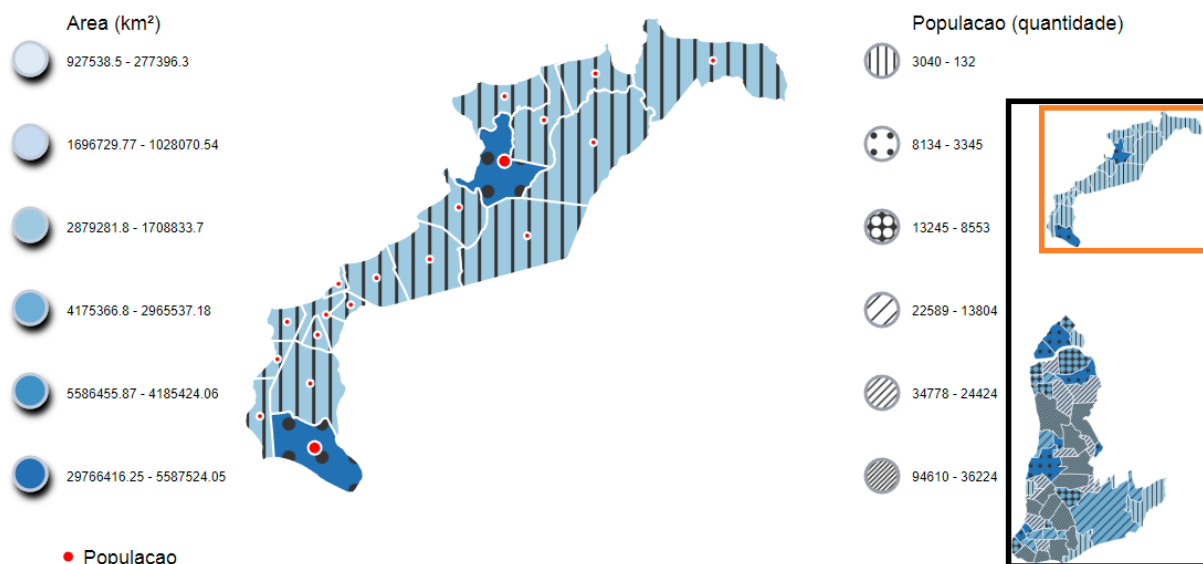


Fonte – O autor

Da mesma forma que o *zoom*, arrastar o mapa também ajuda ao usuário em se concentrar no ponto de interesse do mapa. A Figura 39 exibe o mapa de Belém posicionado um pouco mais para cima, de forma a dar mais foco aos bairros de Mosqueiro.

Figura 39 – Mapa de Belém com foco na área de Mosqueiro.

Nível de Hierarquia: Brasil Para Belem



Fonte – O autor

Os filtros localizados no lado esquerdo da ferramenta permitem remover alguns elementos do mapa. A Figura 40 mostra o mapa do Brasil com os filtros desativados, enquanto a Figura 41 e Figura 42 exibem a atuação dos filtros no mesmo mapa.

Figura 40 – Mapa do Brasil sem filtros ativados.

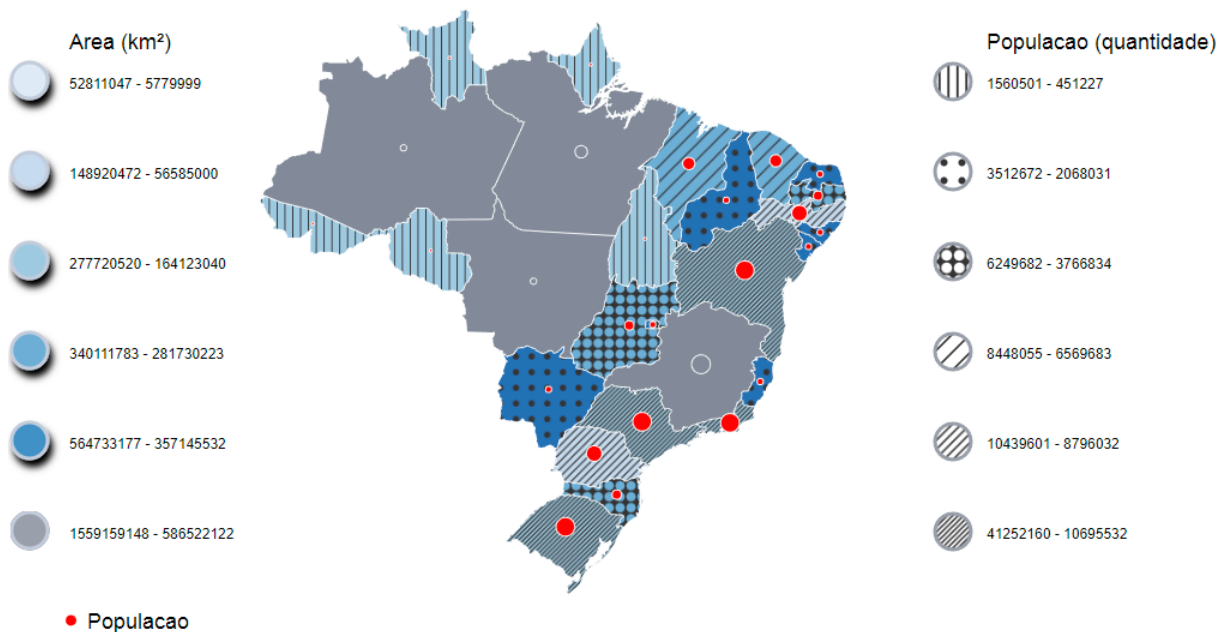
Nível de Hierarquia: Brasil



Fonte – O autor

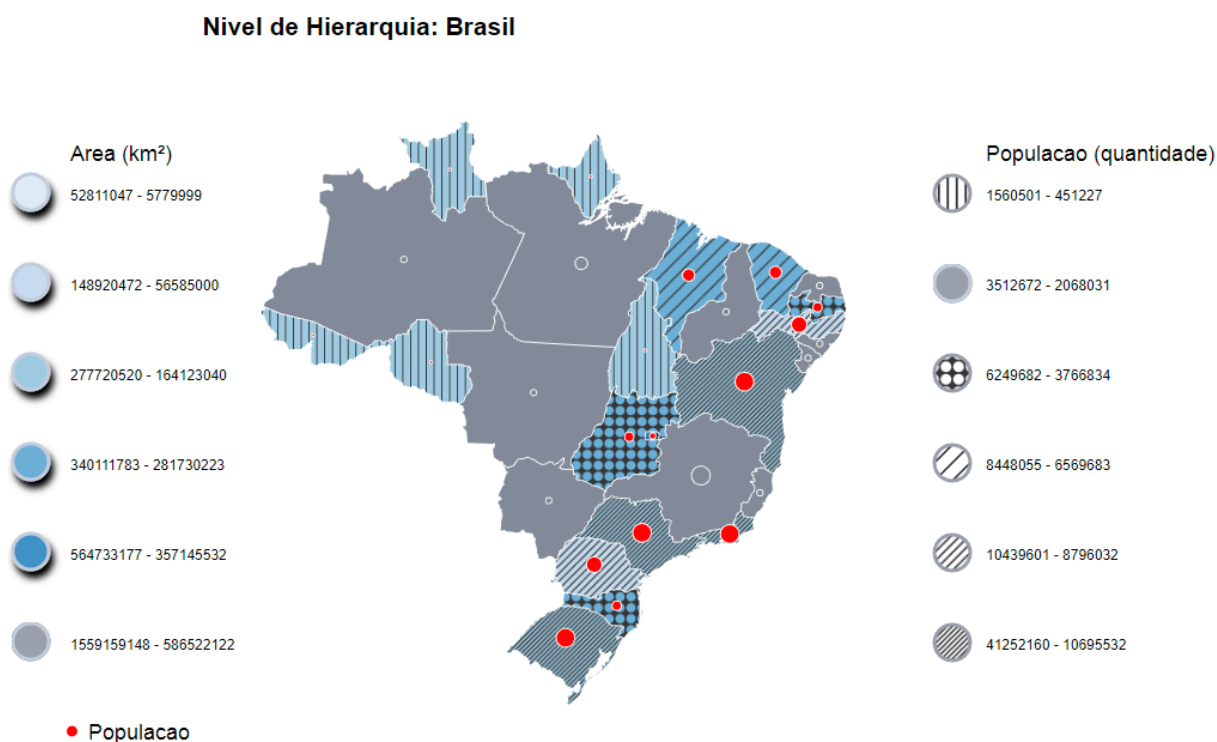
Figura 41 – Mapa do Brasil com um filtro ativado.

Nível de Hierarquia: Brasil



Fonte – O autor

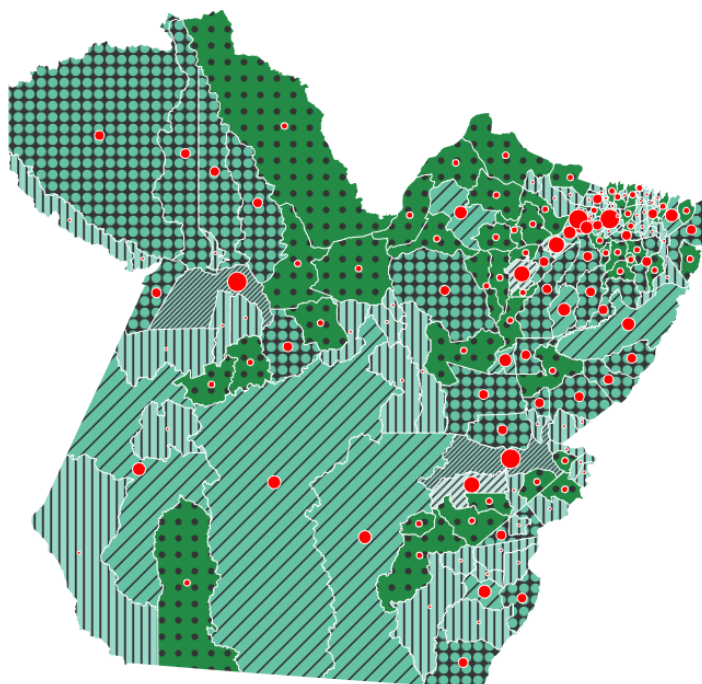
Figura 42 – Mapa do Brasil com dois filtros ativados simultaneamente.



Fonte – O autor

Outras funcionalidades da ferramenta são as de alterar as cores dos elementos e as texturas, as quais podem ser interessantes caso tenha necessidade de enfatizar os dados, como por exemplo, ao exibir dados sobre a temperatura o tema de cores pode ser mais avermelhado com tons laranjas ou uma representação da floresta de um lugar na cor verde, incluindo lugares devastados ou protegidos através de uma textura diferente. A Figura 43 e Figura 44 apresentam essas funcionalidades.

Figura 43 – Exemplo de mudança do tema de cores para verde.



Fonte – O autor

Figura 44 – Exemplo de modificação das texturas, mesclando diferentes formas geométricas.



Fonte – O autor

A ferramenta está disponível para acesso no repositório *Git hub* pessoal do autor¹.

¹ <https://github.com/marcosbenaion/ChoroplethModel>

4 Considerações Finais

Este trabalho realizou uma pesquisa sobre Visualização da Informação aplicada à Visualização Geográfica com foco em mapas *Choropleth*. O problema de dissociação entre dados demográficos e geográficos de Belém foi solucionado com a criação de uma visualização de mapas que mostrou esses dados de forma adequada. A criação da ferramenta foi um processo que envolveu muitas decisões sobre arquitetura, tecnologias e até mesmo sobre os dados.

Uma característica da visualização é a dependência dos dados, o que pode se tornar um ponto fraco quando estes não estão disponíveis. Por outro lado, os pontos fortes do estudo se apresentam com relação a arquitetura flexível, possibilitando adicionar, remover ou modificar o funcionamento dos componentes individuais, alterando as variáveis visuais, por exemplo.

Além disso, a arquitetura proposta divide a aplicação em módulos: Consulta de dados, Gerenciador de Dados, Filtragem, Mapeamento, Renderização e Controle de Interface com o Usuário; cada um realizando uma função específica, que contribuem na criação da visualização de mapas. Sobre esta arquitetura, foram elaborados o diagrama de casos de uso, mostrando as funcionalidades propostas baseadas nas tarefas de visualização, o diagrama de classe, o qual destaca os componentes e suas interações entre si e os diagramas de sequência das principais funcionalidades, detalhando a troca de mensagens entre os componentes.

Nesse sentido, as contribuições realizadas nesse estudo foi a elaboração da arquitetura de uma ferramenta de visualização de mapas *choropleth* hierárquicos e interativos utilizando tecnologias *Web* e a elaboração dos dados geográficos dos bairros de Belém em formato digital. Com este trabalho espera-se que ocorra a difusão sobre o conhecimento das visualizações geográficas e estimule o mapeamento das diferentes regiões do Brasil em arquivos digitais que estejam acessíveis para o público.

4.1 Trabalhos futuros

Os trabalhos futuros para a ferramenta de visualização estão organizados entre melhorias e pesquisas conforme as seguintes listas:

- Melhorias:
 - Armazenar os dados apropriadamente em um Banco de Dados.
 - Possibilitar desativar a exibição de um atributo.

- Permitir outras ações ao selecionar elementos no mapa (através de um menu, por exemplo).
- Adicionar outras variáveis visuais (ou nova combinação delas).
- Permitir a manipulação de outros formatos de arquivos (manipular diretamente arquivos *.svg*, por exemplo).
- Permitir a utilização de outros algoritmos de classificação.

- Pesquisas:

- Adicionar outros tipos de gráficos e analisar as interações ou comportamentos que podem ser inclusos nos mapas (por exemplo menu de contexto, hierarquia exibida graficamente e inserção/alteração em tempo de execução de conjuntos de dados).
- Pesquisar e inserir informações sobre os mapas restantes.
- Estimular a criação dos mapas dos bairros de cada município brasileiro.
- Investigar o máximo de variáveis visuais que podem ser utilizadas simultaneamente e realizar testes com usuários (variáveis visuais únicas e a combinação delas).
- Pesquisar meios de seleção automática do número de classes para classificação baseado nos dados (com o auxílio de histogramas, por exemplo).

Referências

- BOSTOCK, M.; METCALF, C. The topojson format specification. 2013. Disponível em: <<https://github.com/topojson/topojson-specification/blob/master/README.md>>.
- BOSTOCK, M.; OGIEVETSKY, V.; HEER, J. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2301–2309, Dec 2011. ISSN 1077-2626.
- BRAY T., E. The javascript object notation (json) data interchange format. *RFC 7159*, Mar 2014. Disponível em: <<https://www.rfc-editor.org/info/rfc7159>>.
- BUCHIN, K.; SPECKMANN, B.; VERBEEK, K. Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2536–2544, Dec 2011. ISSN 1077-2626.
- BUTLER, H.; DALY, M.; DOYLE, A.; GILLIES, S.; HAGEN, S.; SCHAUB, T. The geojson format. *RFC 7946*, Aug 2016. Disponível em: <<https://www.rfc-editor.org/info/rfc7946>>.
- DEACON, J. Model-view-controller (mvc) architecture. 2005. Disponível em: <<http://sites.google.com/site/urtechfriend/Home/MVC.pdf>>.
- LIU, S.; CUI, W.; WU, Y.; LIU, M. A survey on information visualization: recent advances and challenges. *The Visual Computer*, v. 30, n. 12, p. 1373–1393, Dec 2014. ISSN 1432-2315. Disponível em: <<https://doi.org/10.1007/s00371-013-0892-3>>.
- MACWRIGHT, T. Literate jenkins natural breaks and how the idea of code is lost. 2013. Disponível em: <<https://macwright.org/2013/02/18/literate-jenks.html>>.
- MACWRIGHT, T. Simple statistics. 2016. Disponível em: <<https://simplestatistics.org/>>.
- NETWORK, M. D. About javascript. 2017. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript>.
- PENG, C. Scalable vector graphics (svg). In: *Research Seminar on Interactive Digital Media*. [S.l.: s.n.], 2000.
- RAMOS, C. d. S. *Visualização cartográfica e cartografia multimídia*. [S.l.]: Editora UNESP, 2003. ISBN 8571395950.
- SCALCO, R. Svg patterns for data visualization. 2015. Disponível em: <<http://riccardoscalco.github.io/textures/>>.
- SHNEIDERMAN, B. The eyes have it: a task by data type taxonomy for information visualizations. In: *Proceedings 1996 IEEE Symposium on Visual Languages*. [S.l.: s.n.], 1996. p. 336–343. ISSN 1049-2615.
- SPENCE, R. *Information Visualization: Design for Interaction*. [S.l.]: Pearson, 2007. ISBN 0132065509.

W3C. Html 5.1. 2016. Disponível em: <<https://www.w3.org/TR/html>>.

WARD, M.; GRINSTEIN, G.; KEIM, D. *Interactive Data Visualization: Foundations, Techniques, and Applications*. [S.l.]: CRC Press, 2010. (360 Degree Business). ISBN 9781439865545.