

Visualização de dados pluviométricos com utilização de Banco de Dados não relacional.

Guilherme H. L. dos Santos Milton H. Shimabukuro

Departamento de Matemática e Computação - Universidade Estadual Paulista (UNESP)

19.060-900 - Presidente Prudente - SP - Brasil

g.lourenco.santos@gmail.com, miltonhs@fct.unesp.br

Abstract

In this work, visualization techniques were applied to pluviometric data obtained from measurement stations, it was added interactivity and ways to show the visualizations simultaneously and coordinated, so that more information could be obtained. As support, the javascript library D3.js was used for data-oriented visualization, and the management of the data was done by PostgreSQL, with the JSONB document-oriented NoSQL feature, seeking to better adapt to the data type, using a hybrid approach joining to a relational database with non-relational resources. A WEB environment was built and, through the visualizations, it was possible to extract information about rain averages at various time scales, the consistency periods of the rainfall data of each station, the monthly rainfall behavior of several stations, and the space positioning of the stations. And it was possible to associate this informations through the simultaneous and coordinated presentation of the visualizations. It was also possible to analyze the use of the hybrid approach to data management, and the use of the D3.js library to construct the visualizations.

Resumo

Neste trabalho, técnicas de visualização foram aplicadas a dados pluviométricos obtidos de diversas estações de medição, utilizando interatividade, e disponibilizando visualizações simultâneas e coordenadas, para que mais informações pudessem ser obtidas. Como apoio, foi utilizada a biblioteca javascript D3.js, para visualização orientada a dados, e a gerência dos mesmos foi feita pelo PostgreSQL, com o recurso NoSQL orientado a documento JSONB, buscando melhor se adaptar ao tipo dos dados, utilizando uma abordagem híbrida unindo uma base de dados relacional com recursos não relacionais. Então foi construído um ambiente WEB em que, através das visualizações, foi possível extrair informações sobre as médias de chuva em diversas escalas temporais, o períodos de consistências dos dados pluviométricos de cada estação, o comportamento mensal de chuvas de diversas estações, e o posicionamento espacial das estações, sendo possível associar estas informações através da apresentação simultânea e coordenada das visualizações. Foi possível também analisar o uso da abordagem híbrida para a gerência dos dados, e o uso da biblioteca D3.js para construir as visualizações.

1 Introdução

A análise de dados pluviométricos é de grande importância pois é uma variável dentro de diversos estudos sobre fenômenos da natureza, estando conectada a diversos fatores como a erosão de solos, o enchimento de rios, o comportamento da fauna e flora, e dentre outros, com aplicações em diversas áreas como nos setores agrícolas e agropecuários, de transporte aéreo, rodoviário e naval, de saneamento, e outros (CREPANI; MEDEIROS; PALMEIRA, 2004), (OLIVEIRA; ASSUNÇÃO, 2009), (ASSAD; SANO, 1998). Os dados de chuva são capturados por várias estações de medição na escala diária, e podem ser analisados dia a dia, mês a mês, ano a ano e, eventualmente, analisa-se os dados em todas as escalas ao mesmo tempo. Tais dados também podem ser muito volumosos, e de complexa recuperação dentro de algumas bases de dados.

Devido ao volume de dados e diferentes formas de análise em diferentes escalas de tempo, é importante que exista uma maneira de visualizar vários dados pluviométricos ao mesmo tempo de forma clara e estruturada, conforme alguns trabalhos, tais como o de Branco (2003) que utilizou a visualização para explorar dados pluviométricos, de Romero-Lépiz (2015) que utilizou a visualização para correntes de vento em um ambiente WEB com o uso do D3.js¹ e de Genghini (2011) que incorporou recursos de interatividade a uma plataforma de visualização já existente, é possível observar que a visualização por meio de ferramentas computacionais pode auxiliar nestas análises, como o uso do JavaScript²(linguagem de programação baseada em *scripts*) com a biblioteca D3.js para gerar imagens SVG(*Scalable Vector Graphics*). O volume dos dados também pode influenciar na maneira de armazená-los e recuperá-los para a apresentação e interação, então é importante que haja uma forma adequada para fazê-lo. O uso de banco de dados não relacionais pode ser uma opção que viabilize este processo, pois tem obtido sucesso em diversas aplicações com grande volume de dados como: Facebook³, Google⁴ e Twitter⁵.

Existe também um número grande e crescente de diferentes plataformas de software e de diferentes dispositivos computacionais em uso (CISCO, 2017), o que torna o desenvolvimento de aplicações presentes em todos mais difícil, e com um alto consumo de tempo neste desenvolvimento devido a diferença de sistemas operacionais com diferentes linguagens de programação específicas para cada um (CHMIELEWSKI, 2014). Uma possível solução para este problema seria a construção de aplicações WEB⁶, pois se baseiam em aplicações que utilizam os navegadores⁷ WEB como plataforma. Os navegadores são outro tipo de aplicações que interpretam diversos tipos de documentos e mídias, e estão presente na maioria dos dispositivos (COULOURIS et al., 2013), (BERNERS-LEE, 1991).

Este projeto teve como objetivo apresentar formas de visualização e interação com dados pluviométricos de forma coordenada, buscando facilitar a análise dos mesmos, disponibilizando-as em uma aplicação WEB. Também teve-se como objetivo utilizar uma base de dados para

¹ <https://d3js.org/>, acessado em 7 de Agosto de 2017

² <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>, acessado em 7 de Agosto de 2017

³ <https://www.facebook.com/>, acessado em 15 de Agosto de 2017

⁴ <https://www.google.com.br/>, acessado em 15 de Agosto de 2017

⁵ <https://twitter.com/>, acessado em 15 de Agosto de 2017

⁶ <https://www.w3.org/2008/webapps/>, acessado em 7 de Agosto de 2017

⁷ <https://www.w3.org/TR/webarch/>, acessado em 7 de Agosto de 2017

melhor adaptar o armazenamento e recuperação aos tipos dos dados, aplicando uma abordagem híbrida SQL+NoSQL, utilizando o Sistema Gerenciador de Banco de Dados(SGBD) PostgreSQL+NoSQL com o recurso JSONB, a fim de verificar a complexidade da utilização e flexibilidade na gerência dos dados. O recurso JSONB será melhor explicado na Seção 3.2.

As Seções a seguir deste artigo estão organizadas da seguinte forma: A Seção 2 apresenta a base teórica para as visualizações utilizadas no projeto; A Seção 3 apresenta os conceitos sobre Bases de Dados Não Relacionais abordando também o seu uso dentro do PostgreSQL; Na Seção 4 é apresentada a abordagem utilizada para a construção do projeto; A Seção 5 discute os resultados e testes realizados; Por fim, a Seção 6 encerra com as conclusões e os trabalhos futuros.

2 Visualização e Interação

Segundo Gantz e Reinsel (2012) existe uma quantidade grande e crescente de dados em diversas aplicações e, segundo Shimabukuro (2004), é importante encontrar técnicas para que se possa recuperar informações destes pois, se não houver formas adequadas de processamento dos mesmos, tais dados se tornam potencialmente inúteis. Informações como dados espaciais, observando a distância entre dados, a densidade dos mesmos e o comportamento ao longo do tempo, ajudam a compreender fenômenos de vários domínios, como turismo, negócios, saúde pública e saneamento, meio ambiente e clima. Então, para que se possa analisar tais dados a ponto de obter informações relevantes, que permitam traçar estratégias, são necessárias técnicas tanto para organizar tais dados quanto para apresentá-los.

Técnicas que dão suporte para extrair de forma automática ou semi-automática informações relevantes de grandes volumes de dados são atribuídas à Mineração de Dados(DM, *Data Mining*) área de estudo que pode ser entendida como parte de um processo maior, chamado de Descoberta de Conhecimento em Base de Dados(KDD, *Knowledge Discovery in Databases*) (FAYYAD, 1998), (FAYYAD et al., 1996).

As fases do processo de descoberta de conhecimento podem ser observadas na Figura 1, elas são:

- **Seleção:** É selecionado um conjunto de dados.
- **Pré-processamento:** Realiza-se a remoção de dados estranhos ou corrompidos, remoção de dados faltantes na base e ordenação de acordo com algum atributo.
- **Transformação:** Faz-se um refinamento, diminuindo o número de variáveis a serem consideradas, ou a divisão dos mesmos em partes menores.
- **Mineração:** Consiste na aplicação de operações a fim de reconhecer padrões entre os dados refinados.
- **Interpretação/avaliação:** Os padrões extraídos são avaliados, podendo levar a uma repetição do processo. Caso aceita, as informações estão prontas para serem utilizadas.

A etapa da Mineração de Dados é o mapeamento dos dados referentes a alguma informação, a validade das mesmas está atrelada ao domínio da aplicação e dos objetivos do usuário, segundo

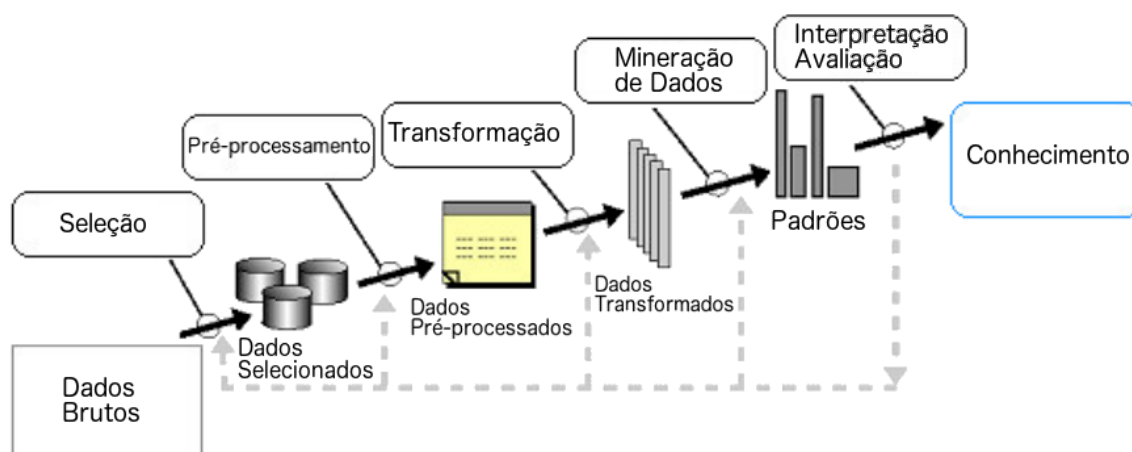


Figura 1 – Passos do processo KDD, adaptado de Fayyad et al. (1996).

Fayyad et al. (1996). Pode ser utilizada para prever novas entradas de dados baseado nos dados já existentes e extrair informações por critérios pré-estabelecidos informando o tipo da informação ou de dado, dentre outros modelos como os estatísticos e de aprendizagem de máquina. Conforme Chen, Han e Yu (1996) são apresentadas categorias de Mineração de Dados baseadas no tipo de informação tomada dos dados:

- **Regras de Associação:** Encontra a frequência em que a presença de um item, em um registro, implica na presença de outros itens no mesmo registro, indicando combinações ou associações. A procura utiliza os parâmetros, onde uma regra de associação “se existe X então existe Y” tem uma confiança ‘c’ e um suporte ‘s’ no qual, a porcentagem de ‘c’ significando o quanto ocorre esta relação e a porcentagem de ‘s’ o quanto X ou Y aparecem nos dados.
- **Classificação de dados:** Os dados são separados em classes, definidas pelos conteúdos de alguns atributos, criadas de um subconjunto de treinamento.
- **Agrupamento (Clustering) de Dados:** Os dados são agrupados, sem uma pré-separação em classes, sendo assim, denominada classificação não supervisionada. Os agrupamentos são feitos dando preferência à similaridades de atributos dentro da classe do que entre as classes, identificando, de acordo com medidas de distância, regiões mais povoadas que são denominadas clusters.
- **Pesquisa por similaridade baseada em padrões:** Pesquisa por atributos similares espaciais e temporais identificando risco, causalidade e tendências associadas ao padrão. Estas informações são importantes para diversas áreas financeiras, meteorológicas e médicas. Tais atributos não precisam ser exatamente idênticos mas ter algum nível de similaridade.

Além da necessidade de extrair informações relevantes de bases de dados, é importante que exista um meio para apresentá-las, para isso são atreladas técnicas de Visualização de Dados, as quais exibem os dados, buscando adquirir mais entendimento e introspecção sobre eles, através de representações visuais como imagens (RHYNE, 2003). Segundo Wong (1999), com o uso

das técnicas conjuntas surge o termo Mineração Visual de Dados (VDM, *Visual Data Mining*), técnicas que buscam unir o refinamento da Mineração de Dados a Técnicas de Visualização com o objetivo de extrair um número ainda maior de informações, aumentando a descoberta de conhecimento, adicionando a capacidade de cognição humana a este processo.

Existem inúmeras técnicas de visualização e interação, este artigo aborda algumas delas: As visualizações de Variação Temporal Uni-escala e Multi-escala (SHIMABUKURO, 2004); *Horizon Graphs* (FEW; EDGE, 2008); Foco+Contexto (NASCIMENTO; FERREIRA, 2005); *Linking and Brushing* (KOSARA; HAUSER; GRESH, 2003).

2.1 Variação Temporal Uni-escala

Shimabukuro (2004) propõe um modelo que pode se aplicar para atributos que apresentam uma variação temporal em uma escala definida (dia, semana, ano, etc) utilizando uma tabela de cores, para mapear os intervalos de valores em interesse nas mesmas. Tais valores podem ser representados por sequências de pixels ou pontos, organizados em células, como na Figura 2. O autor descreve que cada sequência de pixels é correspondente a um objeto espacial, arranjados verticalmente ou horizontalmente. Esta técnica permite uma visão simultânea de vários objetos referentes ao seu valor durante o intervalo de tempo.

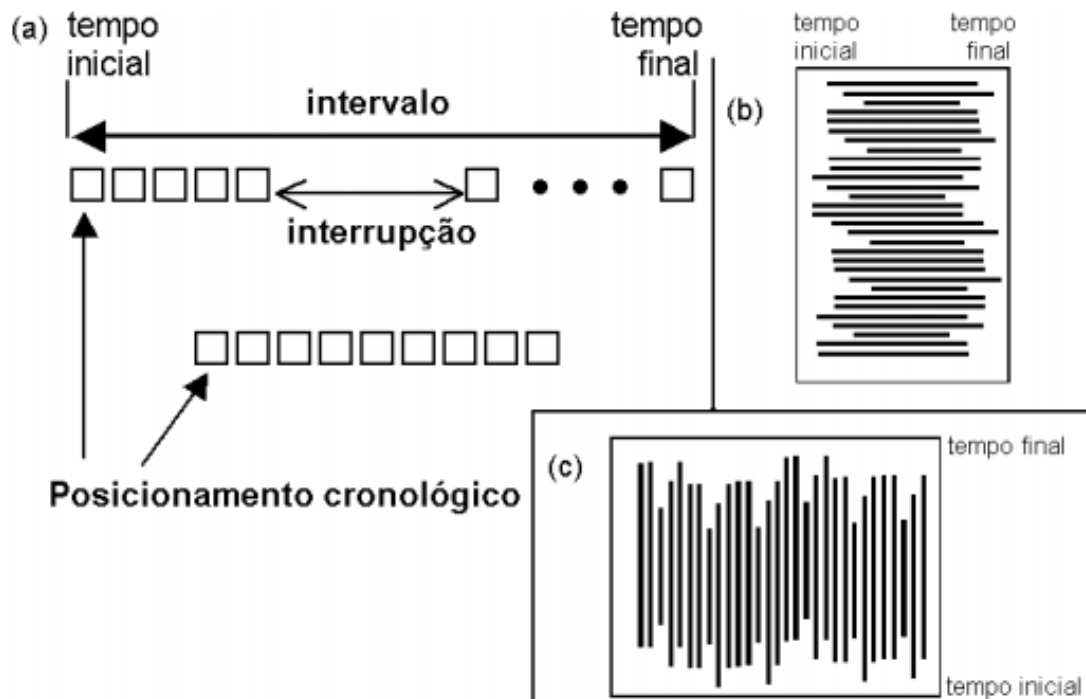


Figura 2 – Construção da representação visual Variação Temporal Uni-escala, (a) exibe o posicionamento das células de forma cronológica presentes dentro de cada faixa que representa os dados de um objeto, (b) exibe a visualização das faixas de objetos com orientação horizontal, (c) exibe a visualização das faixas de objetos com orientação vertical, fonte: (SHIMABUKURO, 2004).

2.2 Variação Temporal Multi-escala

A técnica multi-escala é similar a uni-escala, porém ao invés de ter o foco em observar vários objetos ao longo do tempo, ela tem o foco em observar apenas um objeto, com um detalhamento temporal mais específico. Neste caso, são agrupados os valores do objeto em diferentes escalas de tempo na mesma visualização, permite observar os valores diários, mensais, anuais e no decorrer dos anos de tal objeto como na Figura 3, cada coluna da visualização representa um ano, as primeiras doze células dessa coluna contém de vinte e oito à trinta e uma sub-células cada, contendo o valor diário dos dados, as próximas doze células da coluna contém o valor mensal dos dados e a última célula contém o valor anual dos dados (SHIMABUKURO, 2004).

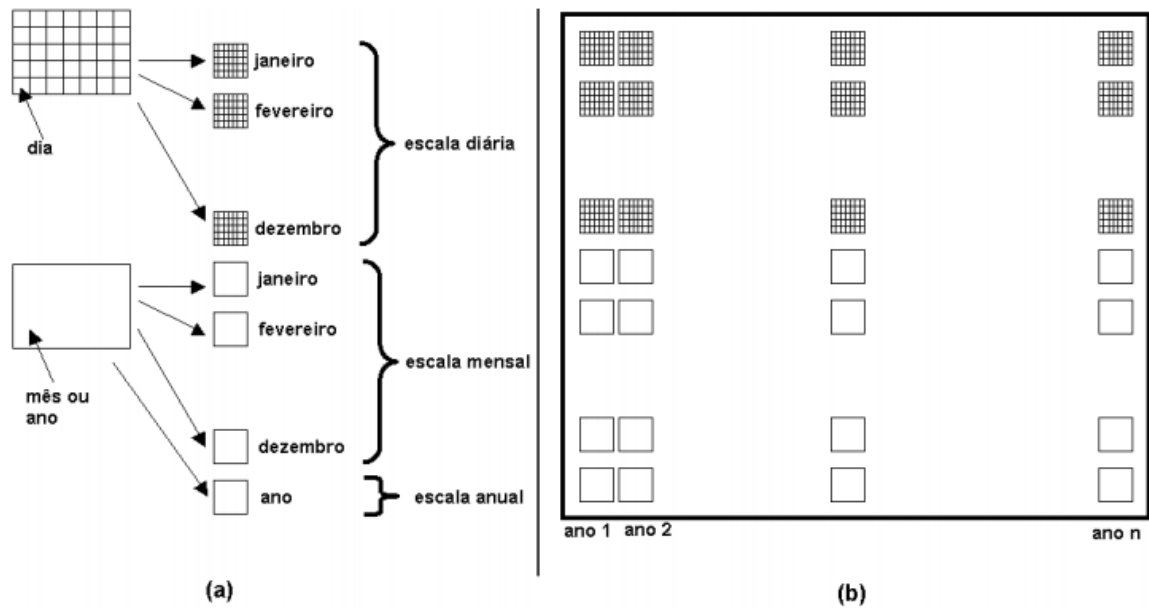


Figura 3 – Construção da representação visual ‘Variação Temporal Multi-escala’, (a) exibe o conteúdo detalhado dentro de cada escala, (b) exibe a visualização com todas as escalas arranjadas juntas, fonte: (SHIMABUKURO, 2004).

2.3 Horizon Graphs

Segundo Perin, Vernier e Fekete (2013) está é uma técnica de visualização que foi primeiro desenvolvida pelo nome de *two-tone pseudo-coloring* (SAITO et al., 2005) e, posteriormente, foi redesenhada e renomeada pela empresa Panopticon. De acordo com Perin, Vernier e Fekete (2013), na construção desta visualização, primeiro os valores de cada dado são plotados em um eixo x,y. O gráfico formado é dividido horizontalmente por um valor base passado como parâmetro, valores acima deste vão ser considerados positivos e coloridos de uma cor, e valores abaixo serão considerados negativos e coloridos com outra cor, como na Figura 4 (a). Então divide-se novamente horizontalmente por um número de faixas também passado como parâmetro. Para cada faixa é adicionada uma intensidade na cor presente iterativamente, como na Figura 4 (b). Os valores negativos são espelhados, como na Figura 4 (c). Por fim as faixas são recolocadas na linha do valor base, como na Figura 4 (d). Este tipo de visualização permite representar

vários gráficos em um espaço reduzido. Esta construção é exibida na Figura 4.

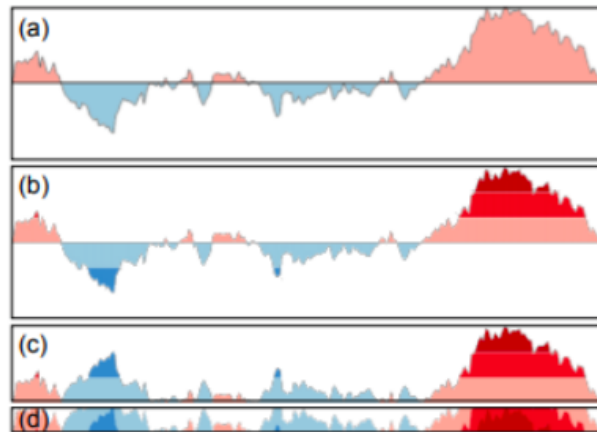


Figura 4 – Construção do Horizon Graph, (a) plot dos dados no eixo x,y e divisão dos valores por valor base com colaração diferente para valores acima e abaixo do valor base, (b) divisão dos valores em faixas com intensidades de cor diferentes, (c) espelhamento dos valores negativos para cima do valor base, (d) reposicionamento das faixas divididas à partir da linha do valor base, fonte: (PERIN; VERNIER; FEKETE, 2013).

2.4 Foco+Contexto

Segundo Nascimento e Ferreira (2005), Lamping, Rao e Pirolli (1995) e Card, Mackinlay e Shneiderman (1999), técnicas deste tipo permitem uma visualização geral dos dados bem como destaque em um grupo dos dados. Permitindo um foco em um conjunto específico de dados sem perder informações dos mesmos no contexto geral. Algumas aplicações dessa técnica são: Perspective Wall, ilustrada na Figura 5, que mostra um calendário onde é possível navegar entre os meses mostrando os três meses consecutivos em foco (MACKINLAY; ROBERTSON; CARD, 1991); Fish-Eye, ilustrada na Figura 6 que exibe os dados com uma escala aplicada na região de interesse (BEDERSON, 2000) e Table Lens, ilustrada na Figura 7, que exibe uma tabela com múltiplo foco em diferentes regiões com sorteio de múltiplas colunas a serem reveladas (RAO; CARD, 1994) .

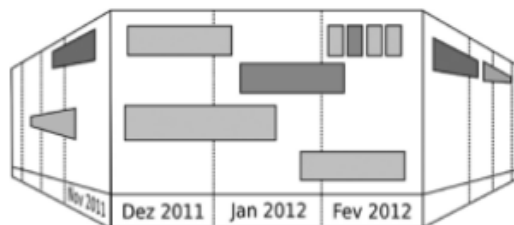


Figura 5 – Perspective Wall aplicado a um calendário, exibe os três meses em foco à frente, nas laterais exibe os meses anteriores e posteriores aos três meses, fonte: (MACKINLAY; ROBERTSON; CARD, 1991).

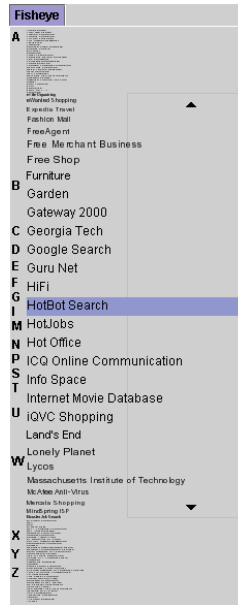


Figura 6 – Fisheye aplicado a uma lista de nomes, os nomes em foco estão com uma escala de aumento gradual do ponto de menor foco ao ponto de maior foco, fonte: (BEDERSON, 2000).

Table Lens: Baseball Player Statistics																												
Calculate:																												
Players	At	Hi	Ho	Ru	Rb	Wa	Ye	Career At	Career Hit	Co	Ca	Es	Ca	Le	01	Pe	Pa	As	Br	Sa	Le	Pe						
Pete Rose								14053	4256																			
Charlie Mo								3926	1029																			
Bob Dernie								1931	491																			
Mike Brown								853	228																			
Tony Armas								28	4																			
Row: 277: Steve Lombardozzi:																												
Column: 10: Career Home Runs:																												
value: 8																												
74 --- ????																												

Figura 7 – Table Lens exibindo três colunas de quatro registros, fonte: (PIROLI; RAO, 2000).

2.5 Linking and Brushing

Esta é uma técnica de coordenação segundo Kosara, Hauser e Gresh (2003), *Brushing* é a capacidade de selecionar valores e regiões de interesse, para serem observados na visualização. *Linking* é a conexão das informações que estão sendo visualizadas em todas as visualizações, quando múltiplas delas estão sendo apresentadas. O uso dessas duas técnicas combinadas permite a seleção de quais conjuntos serão visualizados e todas as visualizações conectadas apresentarão esta informação selecionada no seu próprio modelo. Por serem usualmente implementadas juntas, tornou-se um termo só *Linking and Brushing*. Pode-se ver um exemplo na Figura 8 com múltiplas janelas exibindo dados médicos, compartilhando a mesma cor para os diagnósticos

selecionados para a observação no gráfico, na tabela e no histograma.

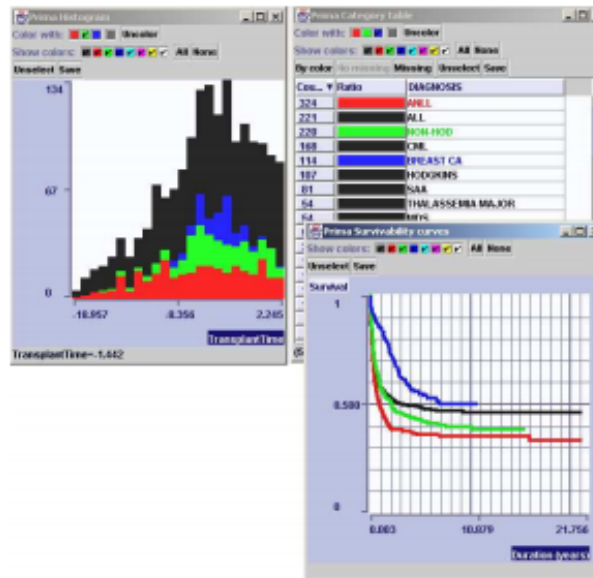


Figura 8 – *Linking and Brushing* aplicado a dados médicos, exibindo um histograma, um gráfico e uma tabela que representam o mesmo conjunto de dados selecionados, fonte: (KOSARA; HAUSER; GRESH, 2003).

3 Banco de Dados Não Relacionais - NoSQL

Conforme Strauch, Sites e Kriha (2011), os bancos de dados relacionais são os tipos predominantes para o armazenamento de dados estruturados, desde 1970. Este tipo de base de dados se suporta em uma aritmética relacional com o proliferado uso do SQL (*Structured Query Language*) para inserir, alterar, recuperar e remover os dados. Bases de dados não relacionais eram utilizadas apenas para aplicações de algum nicho específico como projetos utilizando SOAP⁸ (*Simple Object Access Protocol*) que utiliza o padrão XML⁹ (*Extensible Markup Language*), porém o uso exclusivo do SQL tem sido questionado tanto pelas empresas como pelos acadêmicos devido ao aumento de usuários e volume de dados no meio. Seguindo a argumentação dos autores, para suprir essa demanda algumas alternativas começam a ser propostas, vários outros modelos são criados para abranger mais nichos antes dominados pelos SGBDRs (Sistema Gerenciador de Banco de Dados Relacionais), esses modelos adotam caminhos não relacionais nem sempre garantindo a ACID (Atomicidade, Consistência, Isolamento e Durabilidade) dos dados, porém conseguem otimizar o espaço de armazenamento e a performance das bases de dados, não necessariamente se desfazendo do uso do SQL.

Existem dois pontos de vista não equivalentes sobre o princípio do NoSQL, uma visão apresenta o termo *NoSQL* sendo *No SQL* (Não SQL) que significaria uma base de dados sem SQL (NOSQL... , 2009). Conforme Weber (2010) tem-se outro ponto de vista sendo *NoSQL* como *Not only SQL* (Não apenas SQL), ou seja, base de dados que não seja baseada em SQL porém, pode incluir ou não funcionalidades SQL, podendo existir uma abordagem híbrida. Dentro

⁸ <https://www.w3.org/TR/soap/>, acessado em 7 de Agosto de 2017

⁹ <https://www.w3.org/TR/REC-xml/>, acessado em 7 de Agosto de 2017

destes conceitos existem diferentes tipos de orientação para o armazenamento: Orientado a Colunas, orientado a Grafos, orientado a Documento e armazenamento baseado em Chave-Valor. Na Seção 3.1 será aprofundado o tipo orientado a Documento, pois foi o tipo utilizado neste projeto.

3.1 Armazenamento orientado a Documento

Segundo Weber (2010), este tipo de armazenamento deve ler e escrever em algum tipo conhecido de arquivo e(ou) estrutura. Utiliza documentos com diferentes tipos de conjuntos de dados, alguns tipos de documento são de fácil interpretação para humanos, dentre eles: XML e JSON. O tipo JSON será melhor abordado na Seção 3.2. Também é possível acessar e alterar diretamente tais documentos pelo servidor.

Não existe um esquema pré-definido, o que existe nas bases é a estrutura utilizada no próprio conteúdo do arquivo, sem relação nenhuma dentre os documentos da base, cada documento funciona independente do outro, no caso de bases que os arquivos dependem são chamadas de semi-estruturadas.

Como a estrutura dos arquivos é algo arbitrário, campos sem dados não precisam ser preenchidos, e nenhum espaço de armazenamento precisa ser reservado. Dentre as ferramentas que o implementam estão: CouchDB¹⁰ e MongoDB¹¹. Exemplo de um arquivo em CouchDB na Figura 9, nele está contido um elemento que contém as chaves “_id” e “_rev”, de valores textuais, a chave “yet_another_attribute”, de valor numérico, e a chave “_attachments”, que contém um subelemento como valor, as chaves deste subelemento contém outros subelementos como valor, sendo eles os dados sobre um arquivo de imagem e os dados sobre um arquivo binário.

```
{
  "_id": "a26ab886a7d200a93a32ae192200416d",
  "_rev": "4-c97d88dfa8c79468bcbe03fcf0e5921d",
  "another_attribute": 17,
  "yet_another_attribute": "a_string",
  "_attachments": {
    "Screenshot.png": {
      "content_type": "image/png",
      "revpos": 3,
      "length": 22666,
      "stub": true
    },
    "trace.nam": {
      "content_type": "application/octet-stream",
      "revpos": 2,
      "length": 7830,
      "stub": true
    }
  }
}
```

Figura 9 – Exemplo de um documento JSON em CouchDB, fonte: (WEBER, 2010).

¹⁰ <http://couchdb.apache.org/>, acessado em 13 de Abril de 2016

¹¹ <https://www.mongodb.com/>, acessado em 13 de Abril de 2016

3.2 PostgreSQL e os tipos de dados não relacionais

PostgreSQL é um sistema gerenciador de banco de dados relacional bem estabelecido, com mais de 15 anos de desenvolvimento. Garante ACID e suporta a maioria dos tipos de dados SQL:2008. Tendo também o conjunto de funcionalidades dos SGBDs mais modernos como: triggers, chaves estrangeiras, views e joins. Também tem adotado algumas funcionalidades NoSQL buscando fornecer mais alternativas reunidas em uma base só, evitando que um serviço tenha que usar uma base SQL para parte de sua funcionalidade e outra base NoSQL para outro grupo de funcionalidades, permitindo uma abordagem híbrida, com suporte aos formatos JSON e XML (POSTGRESQL, 1996).

Os tipos JSON e JSONB, que são orientados a Documento baseado na notação JavaScript, são utilizados como forma de armazenamento e troca de dados entre várias plataformas WEB e dispositivos móveis.

JSON *JavaScript Object Notation*, é um formato de dados de fácil leitura e escrita para os humanos, e de fácil geração e análise para as máquinas. Um objeto em JSON é um conjunto de pares do tipo chave-valor que podem ser organizados de forma aninhada, ou seja de forma hierárquica, e(ou) listados em arrays, a chave é uma string e o valor pode ser uma string, um número, um booleano, um array e até mesmo outro objeto aninhado. O array pode ter valores dos mesmos tipos citados. Pode haver também uma lista de objetos no mesmo nível, esta lista pode obedecer ou não a sua ordem (INTERNATIONAL, 2013).

O termo JSONB é a junção de JSON com *Binary*, é a serialização de documentos JSON para binário. Foi desenvolvido para melhorar o desempenho na busca das chaves utilizando sua própria ordenação indexada. Uma grande diferença para o JSON é o uso de tipos de dados com definição de tamanho e tipo pré-determinadas. Para armazenar um número inteiro o JSON utiliza uma cadeia de caracteres arbitrária enquanto o JSONB utiliza 32 ou 64 bits para inteiros e 64 bits para double e floats. O Tipo booleano em JSON é representado pela cadeia de caracteres “True” ou “False”, em JSONB é representado por apenas um bit 0 ou 1. Para valores como cadeias de strings são armazenadas meta- informações como o comprimento dela e, como os outros tipos já tem comprimentos definidos, um parsing na busca por uma chave pode facilmente saltar para posições do conteúdo da mesma dentro do documento (POSTGRESQL, 1996).

Para utilizar os atributos JSON e JSONB foram desenvolvidos novos operadores que podem ser vistos na Tabela 1 e funções auxiliares que podem ser observados na Tabela 2.

Tabela 1 – Operadores de JSON e JSONB, adaptado de (POSTGRESQL, 1996)

Operador	Operando à Direita	Descrição	Exemplo	Resultado do Exemplo
->	int	Retorna o elemento do <i>array</i> JSON	<code>{“a”, “b”, “c”}::json->2</code>	c
->	text	Retorna o objeto da chave	<code>{“a”: [1, 2, 3]}::json->‘a’</code>	[1, 2, 3]
->>	int	Retorna o elemento do <i>array</i> JSON como texto	<code>{‘a’, ‘b’, ‘c’}::json->>1</code>	‘b’
->>	text	Retorna o objeto da chave como texto	<code>{“a”: [1, 2, 3]}::json->>‘a’</code>	‘[1, 2, 3]’
#>	text[]	Retorna o objeto JSON no caminho	<code>{“a”: {“b”: 2}}::json#>‘{a,b}’</code>	2
#>>	text[]	Retorna o objeto JSON como texto no caminho	<code>{“a”: {“b”: 2}}::json#>>‘{a,b}’</code>	2

Tabela 2 – Algumas funções para processar JSON, adaptado de (POSTGRESQL, 1996)

Função	Tipo de Retorno	Descrição	Exemplo	Resultado do Exemplo
json_array_length(json) jsonb_array_length(jsonb)	int	Retorna número de elementos do <i>array</i> JSON	json_array_length('[1, 2, 3, 4, 5]')	5
json_object_keys(json) jsonb_object_keys(jsonb)	Conjunto de texto	Retorna o conjunto de chaves do objeto JSON	json_object_keys('{ "a": 1, "b": 2, "c": 3 }')	('a';'b';'c')
json_array_elements(json) jsonb_array_elements(jsonb)	Conjunto de JSON	Retorna o conjunto de valores do <i>array</i> JSON	json_array_elements('{ { "a": 1 }, { "b": 2 }, { "c": 3 } }')	(({"a": 1}); {"b": 2}; {"c": 3})

Os tipos JSON e JSONB podem ser criados como um atributo qualquer de qualquer tabela, basta identificar o tipo após o nome do campo, como pode ser visto no Código 1.

Código 1 – Criação de uma tabela utilizando atributos JSON e JSONB

```
CREATE TABLE eventos (
    attrJSON JSON,
    attrJSONB JSONB,
    ...
);
```

A inserção também ocorre como em outros atributos de uma tabela, a diferença nesse caso é o conteúdo da coluna, como pode ser visto no Código 2.

Código 2 – Inserção em uma tabela utilizando atributos JSON e JSONB

```
INSERT INTO eventos values(
    '{ "nome": "festa" } '::JSON,
    '{ "listaDeConvidados":
        [ "Joao", "Maria", "Roberto" ]
    } '::JSONB,
    ...
);
```

Pode-se utilizar os operadores e funções dentro do *SELECT* e cláusulas *WHERE*. A consulta abaixo recupera todas as tuplas de eventos com as colunas id e a lista de convidados onde o nome do evento seja festa, como pode ser visto no Código 3.

Código 3 – Consulta em uma tabela utilizando atributos JSON e JSONB

```
SELECT id, attrJSONB->'listaDeConvidados'
FROM eventos
WHERE attrJSON->>'nome' = 'festa'
```

4 Abordagem realizada

A abordagem realizada consistiu na construção de uma aplicação WEB, portanto que pudesse ser acessada por navegadores, para uma visualização interativa de dados pluviométricos já obtidos de estações de medição presentes no estado de São Paulo. Conforme mostra a Figura 10, pode-se observar o uso de diferentes tecnologias para a construção deste ambiente, e que a interação entre a aplicação WEB e os navegadores segue o esquema cliente-servidor, ou

seja, os navegadores, que são os clientes, realizam as requisições e o servidor da aplicação WEB envia as respostas. Para armazenar os dados foi utilizado o SGBD PostgreSQL, com o auxílio de atributos NoSQL utilizando JSONB. O serviço WEB foi feito na linguagem Python com o *framework* Django¹², isto permitiu um encapsulamento para lidar com requisições HTTP (*Hyper Text Transfer Protocol*) e ao lidar com a comunicação no Banco de Dados. O serviço WEB recebe requisições de navegadores e retorna páginas WEB como resposta, ou apenas dados no formato JSON dependendo da requisição. Esta configuração está no lado do servidor. No lado do cliente, os navegadores fazem as requisições ao servidor, e interpretam as páginas WEB que são encaminhadas como resposta. As páginas WEB são construídas com HTML¹³ (*HyperText Markup Language*), CSS¹⁴ (*Cascading Style Sheets*) e JavaScript que são interpretados pelo navegador. Foi utilizada a biblioteca JavaScript D3.js para gerar as visualizações em imagens SVG. A biblioteca utiliza toda a estrutura e o estilo da página para construir as imagens.

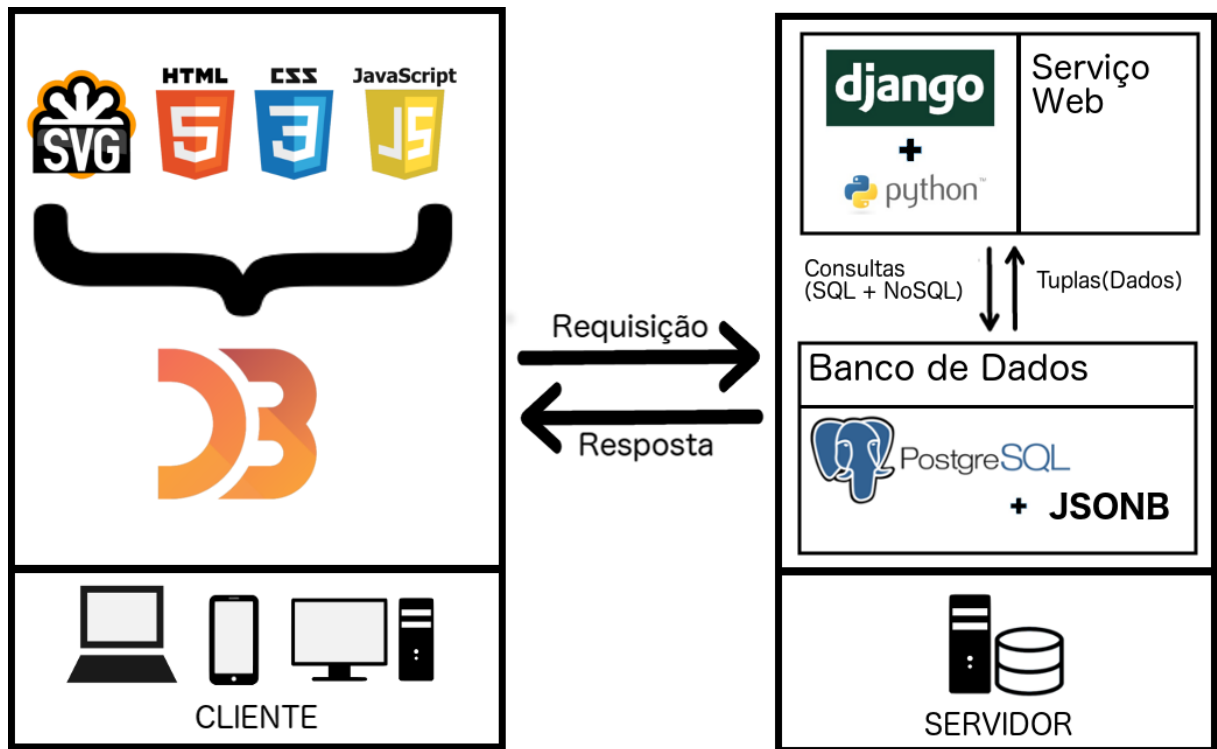


Figura 10 – Estrutura do projeto.

Inicialmente, os dados estavam em arquivos CSV¹⁵ (*Comma-separated values*), os arquivos escolhidos estavam em dois padrões, um para os dados sobre chuvas e o outro para os dados sobre as estações de medição.

Os dados sobre chuvas estavam em arquivos onde o nome do arquivo é o prefixo da estação que foi feita a coleta, o prefixo, além de identificar a estação, identifica a seção e subseção da estação, ele pode ser expresso pela expressão regular: $\text{^[A-Z][0-9]-[0-9][0-9][0-9]\$}$. A letra representa a seção e o primeiro dígito que a acompanha seria a subseção. A primeira linha do

¹² <https://www.djangoproject.com/>, acessado em 7 de Agosto de 2017

¹³ <https://www.w3.org/html/>, acessado em 9 de Agosto de 2017

¹⁴ <https://www.w3.org/TR/REC-CSS2/cover.html>, acessado em 7 de Agosto de 2017

¹⁵ <http://creativyst.com/Doc/Articles/CSV/CSV01.htm>, acessado em 7 de Agosto de 2017

arquivo indica o número de colunas das próximas linhas, a segunda linha identifica o valor de cada coluna, as seguintes linhas contêm os dados separados por colunas ano, mês e quantidade de chuva dos dias, sendo cada linha um mês de um ano, a quantidade 9999 de chuvas representa a inexistência da coleta de dados na data, esta estrutura pode ser observada na Figura 11.

33	Ano,Mes,	1,	2,	3,	4,	5,
1959;01;	0;	0;	20;	102;	192;	
1959;02;	87;	14;	381;	0;	190;	
1959;03;	0;	0;	0;	0;	0;	
1959;04;	0;	0;	0;	0;	0;	

...

26,	27,	28,	29,	30,	31
206;	0;	0;	56;	101;	0
0;	0;	60;	9999;	9999;	9999
0;	0;	0;	77;	0;	0
0;	0;	0;	0;	0;	9999

33						
Ano	Mes	1	2	3	4	5
1959	1	0	0	20	102	192
1959	2	87	14	381	0	190
1959	3	0	0	0	0	0
1959	4	0	0	0	0	0

...

25	26	27	28	29	30	31
51	206	0	0	56	101	0
60	0	0	60	9999	9999	9999
0	0	0	0	77	0	0
54	0	0	0	0	0	9999

Figura 11 – Parte de um arquivo CSV com dados de chuvas em texto plano e formatado em tabela.

O arquivo escolhido para obter informações sobre as estações têm como primeira linha o número de colunas das próximas linhas, a segunda linha identifica o valor de cada coluna, as seguintes linhas contêm, separados por colunas, os dados sobre prefixo, nome, município, bacia hidrográfica, altitude, latitude, longitude, ano Inicial da coleta, ano final da coleta, tamanho do intervalo de anos e consistência, que significa se aqueles dados foram confirmados como reais, cada linha corresponde a uma estação, esta estrutura pode ser observada na Figura 12.

11

Prefixo,Nome,Município,Bacia,Altitude,Latitude,Longitude,Ano Inicial,Ano Final,Intervalo,Consistência

A7-003,ARABA, GUARANI D'OESTE, GRANDE, 440, 1953, 5025, 1970, 1997, 23, 1970/ 1971/ 1972/ 1973/ 1974/ 1975/ 197

A7-004,FAZ. PADUA DINIZ, MIRA ESTRELA, GRANDE, 400, 1954, 5011, 1970, 1997, 21, 1970/ 1972/ 1973/ 1974/ 1975

A7-001,POPULINA, POPULINA, GRANDE, 440, 1956, 5032, 1969, 1997, 20, 1973/ 1974/ 1975/ 1976/ 1977/ 1978/ 1979/ 1

A6-001,RIOLANDIA, RIOLANDIA, GRANDE, 400, 1958, 4941, 1959, 1997, 23, 1970/ 1971/ 1972/ 1973/ 1974/ 1975/ 1976/

A7-002,INDIAPORA, INDIAPORA, GRANDE, 450, 1959, 5015, 1969, 1997, 22, 1971/ 1972/ 1973/ 1974/ 1975/ 1976/ 1977/

11										
Prefixo	Nome	Município	Bacia	Altitude	Latitude	Longitude	Ano Inicial	Ano Final	Intervalo	Consistência
A7-003	ARABA	GUARANI D'OESTE	GRANDE	440	1953	5025	1970	1997	23	1970/ 1971/ 1972/
A7-004	FAZ. PADUA DINIZ	MIRA ESTRELA	GRANDE	400	1954	5011	1970	1997	21	1970/ 1972/ 1973/
A7-001	POPULINA	POPULINA	GRANDE	440	1956	5032	1969	1997	20	1973/ 1974/ 1975/
A6-001	RIOLANDIA	RIOLANDIA	GRANDE	400	1958	4941	1959	1997	23	1970/ 1971/ 1972/
A7-002	INDIAPORA	INDIAPORA	GRANDE	450	1959	5015	1969	1997	22	1971/ 1972/ 1973/

Figura 12 – Parte de um arquivo CSV com dados de estações em texto plano e formatado em tabela.

O conteúdo destes arquivos foi remodelado em tabelas do SGBD PostgreSQL, tem-se três tabelas, uma para Seção, outra para Subseção e uma para a Estação, a configuração das tabelas pode ser observada através do esquema ilustrado na Figura 13.

A tabela Seção contém o id e o nome da seção retirado do prefixo, a tabela Subseção contém



Figura 13 – Esquema construído com SQL e NoSQL(JSONB) para armazenar os dados de chuva e estações.

id, nome da subseção retirado do prefixo e seção que é uma chave estrangeira para Seção, as tabelas Seção e Subseção utilizam uma coluna do tipo *array* para armazenar as coordenadas do retângulo que as delimitam espacialmente, a tabela Estação contém id, subseção, que é uma chave estrangeira para a tabela Subseção, a média de chuvas da estação e todas as colunas restantes do arquivo de dados sobre a estação como atributos.

A média de chuva mensal, anual e da estação foi calculada e adicionada aos dados no momento da inserção dos mesmos à base de dados para não ser necessário processar o cálculo destes em tempo de execução.

Existe um atributo “anos” dentro da tabela Estação do tipo JSONB, ele contém os dados de chuva do arquivo de dados de chuva. Na estrutura deste atributo tem-se um *array* inicial onde cada elemento corresponde a um ano, dentro do elemento ano tem-se o número do ano, a média de chuvas do ano, o número de dias daquele ano, número de dias sem dados e têm-se de forma análoga um *array* de elementos meses e dentro destes um *array* de elementos dias. Assim

tem-se um quadro geral híbrido desta base, a recuperação destes dados é feita de diferentes formas conforme a necessidade de cada visualização, sendo este o resultado do projeto, o uso desta base para as visualizações serão abordados na Seção 5.

5 Testes e análises dos resultados

Tem-se uma apresentação geral do resultado do ambiente na Figura 14, foram utilizadas as visualizações Multi-escala, Uni-escala, Horizon-graph e Foco-contexto, que serão explicadas mais adiante, a visualização Foco+Contexto e o filtro de estações estão sempre visíveis, as visualizações Uni-escala, Multi-escala e Horizon-graph estão divididas em abas que podem ser alternadas.

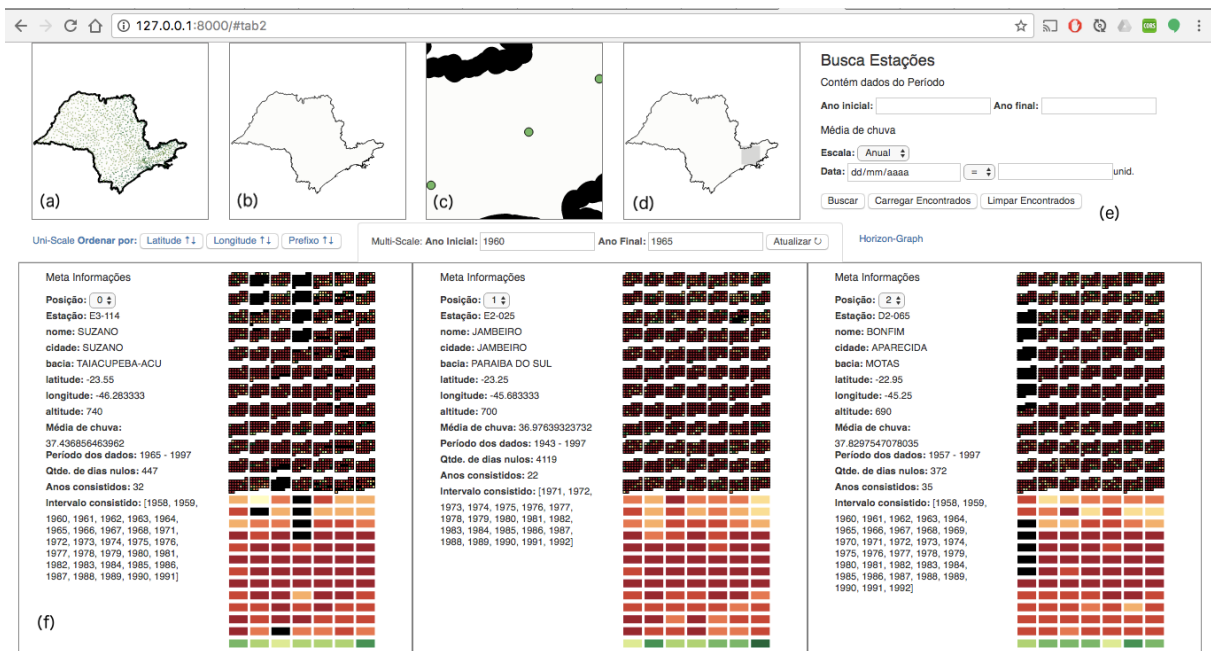


Figura 14 – Aspecto geral do ambiente WEB, (a) mapa com todas as estações, (b) auxiliar para esquema Foco+Contexto de (a), (c) mapa com estações selecionadas, (d) auxiliar para esquema Foco+Contexto de (c), (e) filtro textual de estações, (f) visualizações Multi-escala das estações selecionadas.

A biblioteca D3.js foi utilizada para construir todas as visualizações, facilitando a escrita de código, alguns laços de repetições puderam ser ignorados, pois a biblioteca utiliza os *callbacks* iterativamente à quantidade de elementos dentro dos dados, realizando chamadas implícitas às funções que executam o desenho, dados complexos em JSON puderam ser processados em todas as camadas de subelementos, e obteve-se uma flexibilidade na geração das visualizações, sendo elas adaptáveis ao conteúdo dos dados passados.

O propósito de cada visualização e sua forma de interação serão apresentados nas próximas seções, em adicional ao conjunto de visualizações foi introduzido um filtro de estações por parâmetros textuais, o filtro utiliza uma busca híbrida com parâmetros em SQL e NoSQL na mesma consulta, este recurso também será aprofundado nas seções seguintes.

Além da interatividade presente em cada visualização, foi utilizada a técnica de coordenação

Linking and Brushing, pela qual é possível, de diferentes fontes de visualizações, adicionar uma estação ao conjunto de estações selecionadas para a visualização. Pode-se adicionar a partir da visualização Uni-escala, da Foco-contexto e do filtro SQL+NoSQ, também é possível visualizar as várias estações selecionadas em várias visualizações simultaneamente.

5.1 Visualização Foco+Contexto

No topo do ambiente à esquerda, tem-se quatro mapas do estado de São Paulo, onde espacialmente estão contidas as estações em que foram obtidas as medições pluviométricas. Os círculos representam o local das estações de medição. A cor interna dos círculos representa a média de chuvas de todos os anos da estação. Estas visualizações estão presentes para poder-se associar as informações espaciais que obtem-se delas às outras visualizações. Para esta visualização, são recuperadas todas as estações da base de dados mas com apenas os atributos que serão utilizados, eliminando principalmente o atributo anos, pois é o atributo com a maior quantidade de dados.

Na Figura 14 (a), tem-se o primeiro mapa exibindo todas as estações representadas por círculos, nele são disponibilizadas algumas interações. É possível executar o efeito zoom e pan ilustrado na Figura 15.

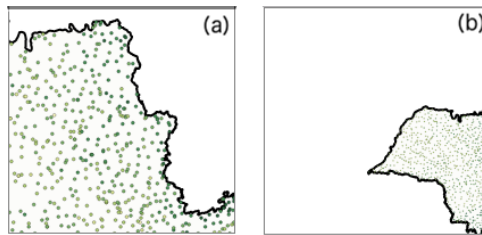


Figura 15 – (a) interação de zoom aplicado na Figura 14 (a), (b) interação de pan aplicado na Figura 14 (a).

O clique em um dos círculos adiciona a estação, representada pelo círculo, ao conjunto de estações selecionadas para visualização e, para facilitar o acerto do clique, foi implementada uma escala de aumento do tamanho no círculo em que o mouse estiver sobre ele, este recurso pode ser visto na Figura 16.

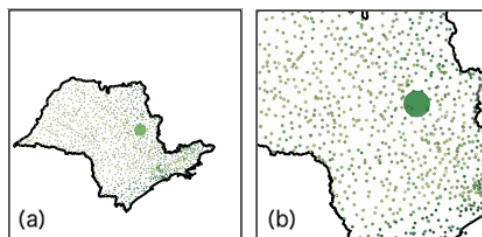


Figura 16 – (a) aplicação de aumento de escala em estação sobre o mouse para facilitar o clique na Figura 14 (a), (b) aplicação de aumento de escala em estação sobre o mouse para facilitar o clique com interação de zoom aplicadas na Figura 14 (a)

Na Figura 14 (b), tem-se o segundo mapa que é parte da visualização Foco+Contexto referente ao primeiro mapa em Figura 14 (a). É exibido o mapa completo destacando a parte visível

do primeiro mapa em um retângulo com pouca opacidade. Esta visualização permite navegar no primeiro mapa através de zoom e pan sem perder a informação de onde se encontra aquela região no mapa completo. No próprio mapa da Figura 14 (b), há a interatividade de desenhar o retângulo que será a visualização do mapa da Figura 14 (a). Este comportamento pode ser observado na Figura 17.

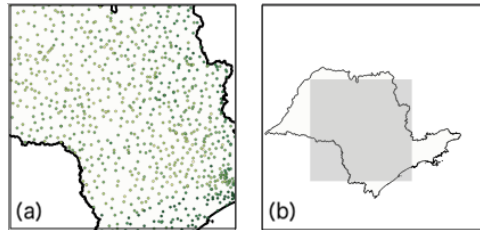


Figura 17 – (a) interação de zoom aplicado na Figura 14 (a), (b) Figura 14 (b) exibindo a área visível no mapa completo.

Na Figura 14 (c), tem-se o terceiro mapa exibindo as estações que foram selecionadas. Neste mapa é calculado o retângulo menor possível que contenha as estações selecionadas, após o cálculo deste retângulo é feito o maior zoom possível na região delimitada do retângulo e, evido a este cálculo e zoom a interação de pan e zoom manual estão desabilitadas neste mapa. O quarto mapa, exibido na Figura 14 (d), segue como parte da visualização Foco+Contexto para dar apoio a visualização do terceiro mapa.

5.2 Visualização Uni-escala

A visualização Uni-escala representa o período em que os dados estão consistidos de cada estação, o prefixo da estação a identifica, é possível adicionar a estação ao conjunto de estações selecionadas clicando no prefixo, e a lista pode ser ordenada por latitude, longitude e prefixo. O objetivo da visualização é permitir uma análise de quais estações possuem dados consistentes dentro do mesmo período para que possam ser comparadas. Os mesmos dados que foram recuperados na visualização Foco+Contexto são reaproveitados nesta, a visualização pode ser observada na Figura 18.

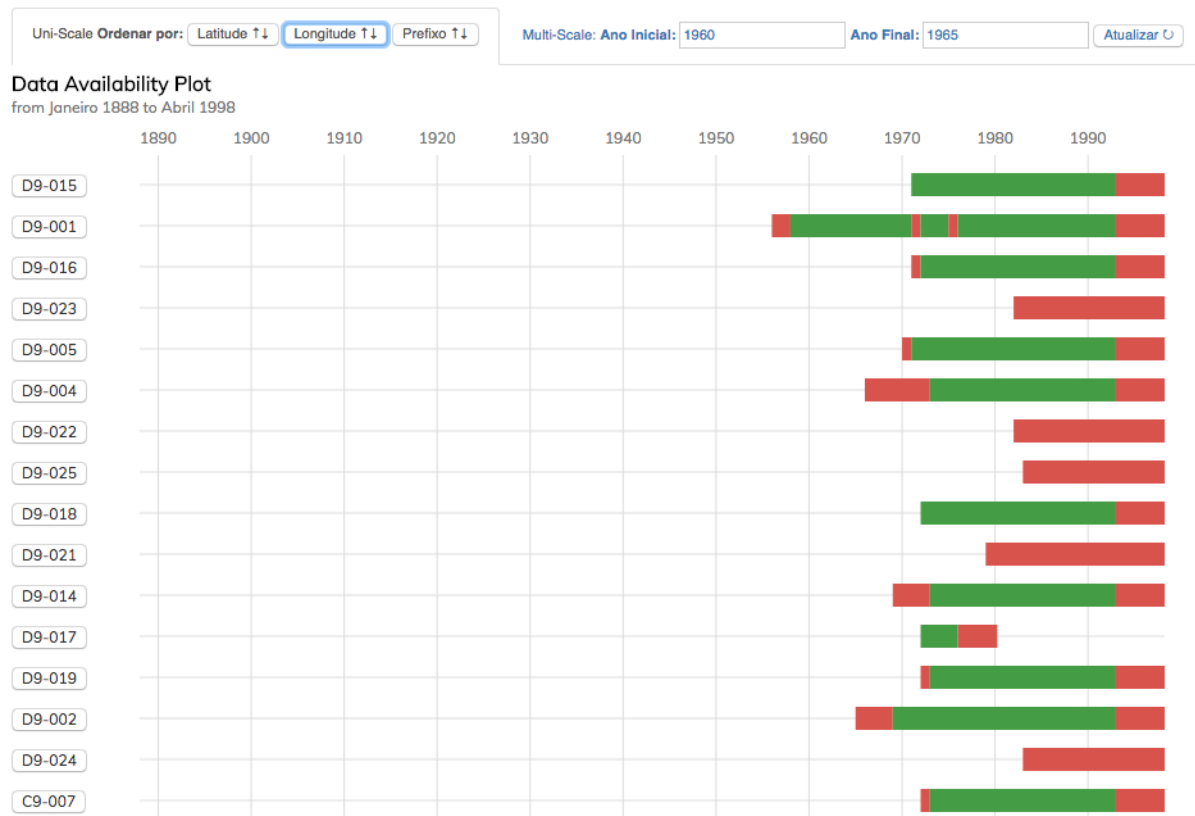


Figura 18 – Visualização Uni-escala informando a consistência dos dados dentro da escala anual.

5.3 Visualização Multi-escala

A visualização Multi-escala exibe a quantidade de chuva durante diferentes escalas de tempo com alguns atributos textuais sobre as estações selecionadas. Nesta visualização, a cor preta significa a inexistência de dados sobre o período, e as outras cores estão dentro de uma escala entre vermelho e verde, onde cores próximas do vermelho significam valores baixos e cores próximas do verde significam valores altos. Várias estações podem ser exibidas e serão apresentadas lado a lado. Os anos que serão exibidos são editáveis na própria aba, permitindo atualizar os dados que estão sendo visualizados, como na Figura 19

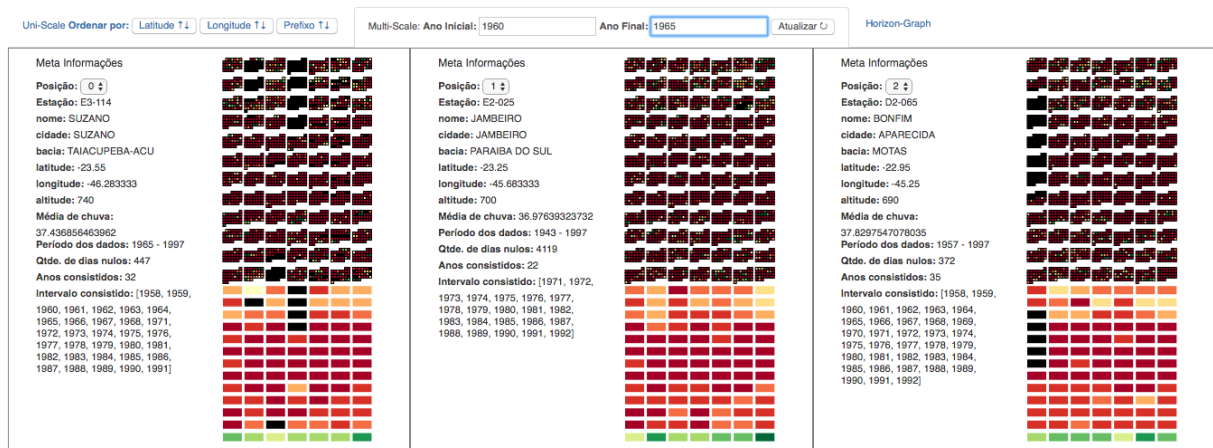


Figura 19 – Visualização Multi-escala com seleção de intervalo de visualização.

É possível visualizar a quantidade de chuva de maneira textual ao passar o mouse por cima das células da visualização, esta interação pode ser observada na Figura 20.

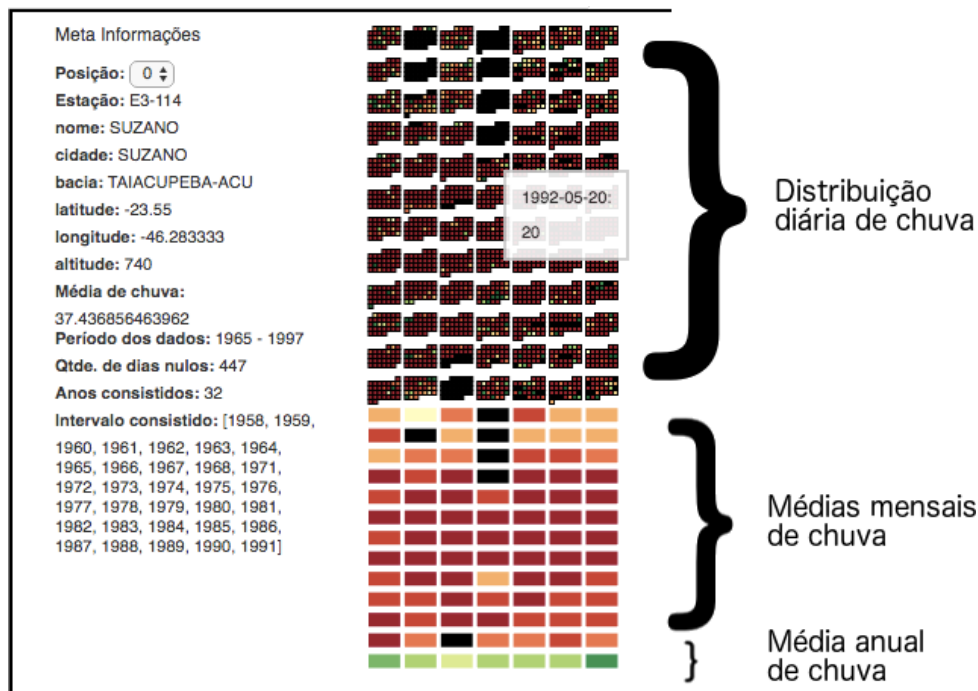


Figura 20 – Informação textual da quantidade de chuva da célula sobre o mouse.

Também existe a possibilidade de alterar a posição das visualizações, permitindo escolher quais estações serão visualizadas com maior proximidade visual, esta funcionalidade pode ser vista na Figura 21.

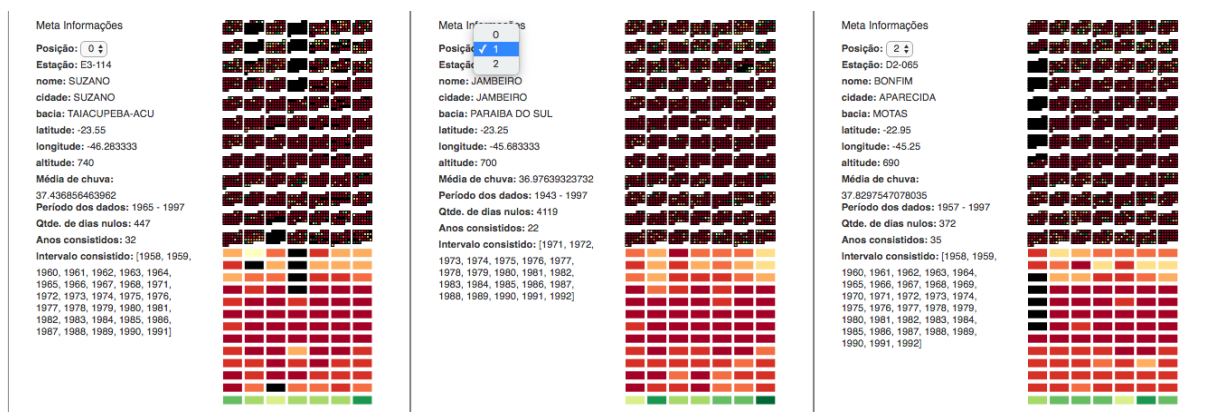


Figura 21 – Realocando a posição das visualizações Multi-escala.

Essa visualização dá suporte para a obtenção de informações de quantidade de chuvas durante longo período de tempo sem perder micro-informações de pequenas escalas de tempo como de dias, podendo também comparar o comportamento de diferentes estações. Para gerar a visualização são recuperados todos os atributos da estação selecionada, no atributo anos ocorre uma filtragem para não carregar anos fora do intervalo a ser visualizado.

5.4 Visualização *Horizon Graph*

A visualização *Horizon Graph* exibe os dados sobre a média de chuva mensal, durante o período selecionado na Multi-escala, das estações selecionadas. O foco desta visualização é exibir várias estações em um espaço verticalmente reduzido e o comportamento das chuvas ao decorrer do tempo. Os dados recuperados na visualização Multi-escala são reaproveitados nesta. No caso de dados pluviométricos não tem-se valor negativo de chuvas, e nenhum valor foi passado como valor base, logo a aplicação do *Horizon Graph* não se divide em cores positivas e negativas. A visualização pode ser observada na Figura 22.

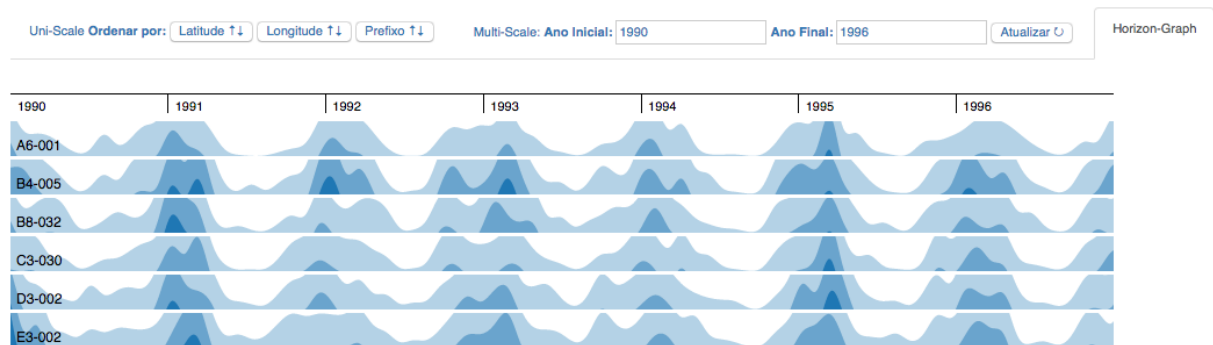


Figura 22 – *Horizon Graph* exibindo a média de chuva mensal durante os anos.

5.5 Filtro de estações

No Filtro de estações é possível realizar a busca e seleção de estações, pode-se filtrar por estações em que: Contenha dados dentro do período do ano inicial ao ano final, a quantidade de chuva da escala da data seja maior, menor, maior ou igual, menor ou igual, ou igual a quantidade inserida. A busca identifica o número de estações encontradas, então é possível adicioná-las ao conjunto de estações selecionadas. O filtro pode ser observado na Figura 23.

The image shows a web form titled 'Busca Estações'. It contains the following elements: a label 'Contém dados do Período', two input fields for 'Ano inicial' and 'Ano final', a label 'Média de chuva', a dropdown menu for 'Escala' with 'Anual' selected, a 'Data' input field with a format 'dd/mm/aaaa', a comparison operator dropdown (currently showing '=') and a unit input field labeled 'unid.'. At the bottom, there are three buttons: 'Buscar', 'Carregar Encontrados', and 'Limpar Encontrados'.

Figura 23 – Filtro para executar buscas textuais por estações.

Para poder levar os resultados em todas as visualizações esse filtro carrega todos os atributos das estações filtradas. Dentre os parâmetros utilizados na filtragem tem-se os atributos SQL: data de início da coleta, data final da coleta. Estes podem ser combinados com a média de chuva presente dentro do atributo NoSQL anos. A média de chuva corresponde ao valor presente na data escolhida dentro da escala anual, mensal ou diária, exemplo: Se escolhida a data 10/07/2017 e a escala mensal, a consulta buscará a média de chuva do mês de Julho do ano de 2017 de todas

as estações. Os operadores de comparação $>$, $<$, $>=$, $<=$, $=$ são diretamente utilizados na comparação do valor da média dentro da consulta. Um exemplo da consulta híbrida pode ser observado no Código 4, esta consulta retorna as tuplas em que os dados estão consistidos entre o período de 1990 a 1996 e quando a média de chuva da data de sete de Julho de 1993 for maior do que quarenta.

Código 4 – Consulta híbrida com cláusulas WHERE verificando atributos SQL e NOSQL

```
WITH years_data AS (
    SELECT id, year_ini, year_end, st_years, prefix, code, ...
    FROM rainfall_station st,
         jsonb_array_elements(st.years) st_years
    WHERE (st_years->>'year')::int = 1993
)
SELECT DISTINCT ON (id) id, year_ini, year_end, prefix, code, ...
FROM years_data WHERE
(
    st_years->'months'->6->'days'->6->>'day_average'
)::float > 40.0
AND year_ini <= 1990 AND year_end >= 1996
```

5.6 Comparação de SQL e NoSQL

Além de observar o resultado das visualizações, é interessante atentar para os diferenciais de uma arquitetura híbrida, podem existir diferentes formas de modelar os dados utilizados neste projeto apenas em SQL mas, para fins de comparação, foi construída somente uma modelagem apenas em SQL conforme o esquema da Figura 24.

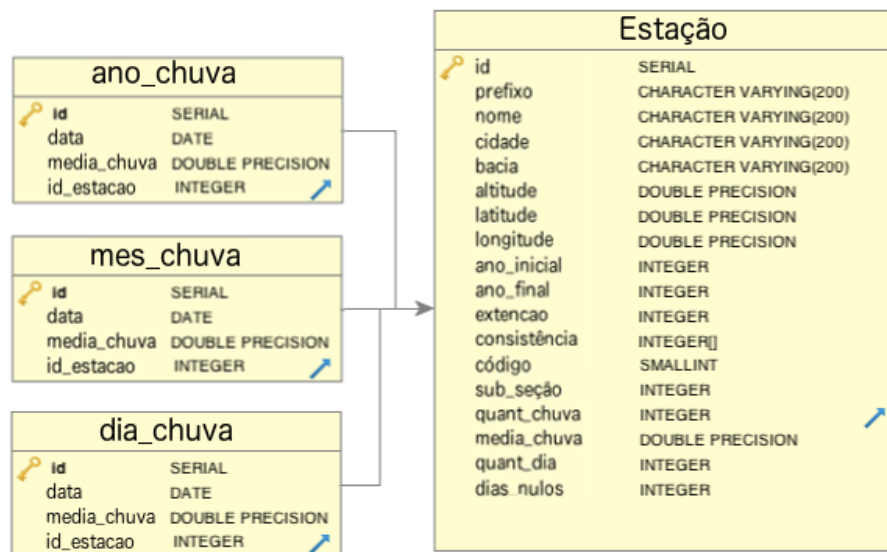


Figura 24 – Esquema modelado em SQL para armazenar os dados de chuva.

Observa-se que neste modelo apenas SQL, para encontrar o valor de chuva de um dia tem-se que primeiro pesquisar as tuplas de `dia_chuva` que pertencem à estação. No modelo deste projeto, em razão do atributo JSONB `anos` já estar pertencente à tupla e, os dias e meses estarem indexados, tem-se um acesso direto aos dias e meses dos anos da estação. Considerando somente a quantidade de tuplas acessadas usando apenas SQL em comparação com a abordagem híbrida, com o objetivo apenas de recuperar os dados de um dia de uma estação em uma determinada data, tem-se algumas observações:

- No caso de apenas SQL, é necessário encontrar todas as tuplas de `dia_chuva` com a chave estrangeira igual ao id da estação, como o acesso a chaves estrangeiras pode ser indexado, é possível desconsiderar este passo, então essa pesquisa percorre, no pior caso, o número de tuplas igual ao número de anos em cada estação aproximadamente, vezes 365 dias aproximadamente, avaliando que tem-se, em média, 20 anos em cada estação, o resultado seria aproximadamente 7.300 tuplas para encontrar a tupla de `dia_chuva` com o atributo id da estação na determinada data.
- No caso do SQL com JSONB, basta verificar o id da tupla da estação desejada dentre todas as tuplas de estações, como o acesso a chaves primárias é indexado pode-se desconsiderar este passo, após encontrar a tupla, basta procurar o ano dentro do atributo `anos`, em média 20 elementos de um *array*, acessar diretamente o mês pelo índice e, posteriormente, o dia pelo índice, desconsiderando o acesso indexado dentro do JSONB, teria apenas 20 tuplas referente ao número médio de anos mas, ainda que fossem considerados os passos indexados dentro do JSONB como se fosse o custo de acesso a tuplas na contagem, teria-se 20 anos aproximadamente dentro da estação já encontrada vezes 2 acessos diretos indexados para meses e dias, seria então 40 itens.

É importante frisar que, embora possa existir alguma outra opção de modelagem apenas em SQL para os dados utilizados no projeto, e que seja tão eficiente ou mais à modelagem híbrida, ainda pode-se analisar este tratamento de forma didática e análoga para cenários em que alguns atributos das tabelas tem valores ordenados mas que não podem ser indexados comumente em bases apenas SQL, ocasionando uma grande quantidade de tuplas para serem processadas. Além de que a pesquisa de tuplas dentro de atributos JSONB facilitou o acesso aos dados por ser possível acessá-los de chaves ou índices dentro do json ao invés de realizar *subqueries*.

Uma dificuldade encontrada foi o tamanho dos dados por tupla pois, todos os dados de chuva sobre a estação estão em apenas um atributo JSONB dentro da tupla, então um recurso essencial para diminuir este problema foi realizar as buscas sem selecionar o atributo JSONB. Apenas utilizá-lo dentro das cláusulas *WHERE* e em casos onde é necessário a recuperação dos mesmos para as visualizações, como na visualização Multi-escala, onde ela utiliza os dados dentro do atributo JSONB, mas mesmo assim, após a recuperação dos dados, ainda é feito um corte deixando apenas os que serão utilizados na visualização.

6 Conclusões e trabalhos futuros

Este trabalho apresentou um ambiente WEB para visualizar dados pluviométricos de forma interativa e coordenada, para isso foram utilizadas várias técnicas de visualização e várias tecnologias para dar apoio ao desenvolvimento, como o uso da biblioteca JavaScript D3.js para gerar as visualizações e aplicando uma abordagem híbrida para o armazenamento e recuperação dos dados com o PostgreSQL+NoSQL.

O uso das técnicas de visualização combinadas com interação e coordenação permitiu observar diferentes informações dos dados em um mesmo contexto, e o uso de banco de dados PostgreSQL+NoSQL foi uma alternativa para solucionar os problemas de recuperação de dados dentro do grande volume utilizado neste projeto, podendo observar os benefícios e dificuldades do uso da abordagem híbrida, além da possibilidade de aplicá-la em outros contextos de aplicações de forma análoga.

Para a continuação deste trabalho sugere-se realizar testes de usabilidade com especialistas da área de climatologia para encontrar possíveis necessidades de adaptar e(ou) adicionar visualizações, como utilizar um período para exibir a média de chuvas da estação nos mapas, e seria interessante também que fosse disponibilizado no ambiente um acesso para atualizar os dados que no momento são apenas recuperáveis.

Referências

- ASSAD, E. D.; SANO, E. E. Sistema de informações geográficas: aplicações na agricultura. EMBRAPA-CPAC, 1998.
- BEDERSON, B. B. Fisheye menus. In: ACM. *Proceedings of the 13th annual ACM symposium on User interface software and technology*. [S.l.], 2000. p. 217–225.
- BERNERS-LEE, T. The original http as defined in 1991. *World Wide Web Consortium (W3C)*, 1991.
- BRANCO, V. M. A. *Visualização como suporte à exploração de uma base de dados pluviométricos*. Dissertação (Mestrado) — Universidade de São Paulo, 2003.
- CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. *Readings in information visualization: using vision to think*. [S.l.]: Morgan Kaufmann, 1999.
- CHEN, M.-S.; HAN, J.; YU, P. S. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, IEEE, v. 8, n. 6, p. 866–883, 1996.
- CHMIELEWSKI, J. Device-independent architecture for ubiquitous applications. *Personal and Ubiquitous Computing*, Springer, v. 18, n. 2, p. 481–488, 2014.
- CISCO. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper*. 2017. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>, acessado em 15 de Agosto de 2017. Acesso em: 2017-08-15.
- COULOURIS, G. et al. *Sistemas Distribuídos: Conceitos e Projeto*. [S.l.]: Bookman Editora, 2013.

- CREPANI, E.; MEDEIROS, J. d.; PALMEIRA, A. F. Intensidade pluviométrica: uma maneira de tratar dados pluviométricos para análise da vulnerabilidade de paisagens à perda de solo. *São José dos Campos: INPE*, 2004.
- FAYYAD, U. M. Mining databases: Towards algorithms for knowledge discovery. *IEEE Data Eng. Bull.*, v. 21, n. 1, p. 39–48, 1998.
- FAYYAD, U. M. et al. Knowledge discovery and data mining: towards a unifying framework. In: *KDD*. [S.l.: s.n.], 1996. v. 96, p. 82–88.
- FEW, S.; EDGE, P. Time on the horizon. *Perceptual Edge: Visual Business Intelligence Newsletter*, p. 1–7, 2008.
- GANTZ, J.; REINSEL, D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, v. 2007, n. 2012, p. 1–16, 2012.
- GENGHINI, F. *Extensão da plataforma de visualização INFOVIS incorporando recursos de interatividade*. Dissertação (Trabalho de Conclusão de Curso) — Universidade Estadual Paulista (UNESP), Campus de Presidente Prudente, 2011.
- INTERNATIONAL, E. *Introducing JSON*. 2013. Disponível em: <http://www.json.org/>.
- KOSARA, R.; HAUSER, H.; GRESH, D. L. An interaction view on information visualization. *State-of-the-Art Report. Proceedings of EUROGRAPHICS*, p. 123–137, 2003.
- LAMPING, J.; RAO, R.; PIROLI, P. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 1995. p. 401–408.
- MACKINLAY, J. D.; ROBERTSON, G. G.; CARD, S. K. The perspective wall: Detail and context smoothly integrated. In: ACM. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 1991. p. 173–176.
- NASCIMENTO, H. A. D.; FERREIRA, C. B. Visualização de informações—uma abordagem prática. In: *XXV Congresso da Sociedade Brasileira de Computação, XXIV JAI. UNISINOS, S. Leopoldo-RS*. [S.l.: s.n.], 2005.
- NOSQL Databases. 2009. Disponível em: <http://nosql-database.org/>. Acesso em: 2016-03-15.
- OLIVEIRA, A. G.; ASSUNÇÃO, W. L. A importância dos dados das variáveis climáticas nas pesquisas em geografia: um estudo de caso empregando a precipitação pluviométrica. *Caminhos de Geografia*, v. 10, n. 31, 2009.
- PERIN, C.; VERNIER, F.; FEKETE, J.-D. Interactive horizon graphs: Improving the compact visualization of multiple time series. In: ACM. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [S.l.], 2013. p. 3217–3226.
- PIROLI, P.; RAO, R. Table lens as a tool for making sense of data. 11 2000.
- POSTGRESQL. *JSON Functions and Operators*. 1996. Disponível em: <https://www.postgresql.org/docs/9.6/static/functions-json.html>.
- RAO, R.; CARD, S. K. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In: ACM. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 1994. p. 318–322.

RHYNE, T.-M. Does the difference between information and scientific visualization really matter? *IEEE Computer Graphics and Applications*, IEEE Computer Society Press, v. 23, n. 3, p. 6–8, 2003.

ROMERO-LÉPIZ, J. A. Herramienta en ambiente web para la visualización de las corrientes de viento en costa rica. Instituto Tecnológico de Costa Rica, 2015.

SAITO, T. et al. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In: IEEE. *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. [S.l.], 2005. p. 173–180.

SHIMABUKURO, M. H. *Visualizações temporais em uma plataforma de software extensível e adaptável*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação (ICMC-USP), São Carlos, 2004.

STRAUCH, C.; SITES, U.-L. S.; KRIHA, W. Nosql databases. *Lecture Notes, Stuttgart Media University*, 2011.

WEBER, S. Nosql databases. *University of Applied Sciences HTW Chur, Switzerland*, 2010.

WONG, P. C. Visual data mining. *IEEE Computer Graphics and Applications*, IEEE, v. 19, n. 5, p. 20–21, 1999.