

Wancharle Sebastião Quirino

SEARCHLIGHT: Facilitando a visualização de informações crowdsourcing em Mapas Web através de Zoom

Vitória - ES, Brasil

XX de maio de 2013

Wancharle Sebastião Quirino

SEARCHLIGHT: Facilitando a visualização de informações crowdsourcing em Mapas Web através de Zoom

Monografia apresentada para obtenção do
Grau de Bacharel em Engenharia de Computação
pela Universidade Federal do Espírito Santo.

Orientador: Celso Alberto Saibel Santos

Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática

Vitória - ES, Brasil

XX de maio de 2013

Wancharle Sebastião Quirino

SEARCHLIGHT: Facilitando a visualização de informações crowdsourcing em Mapas Web através de Zoom

Monografia apresentada para obtenção do
Grau de Bacharel em Engenharia de Compu-
tação pela Universidade Federal do Espírito
Santo.

Trabalho aprovado. Vitória - ES, Brasil, 24 de novembro de 2012:

Celso Alberto Saibel Santos
Orientador

Professor
Convidado 1

Professor
Convidado 2

Vitória - ES, Brasil
XX de maio de 2013

*Este trabalho é dedicado aos meus pais, que sempre estiveram comigo,
por toda a sua dedicação.*

Lista de ilustrações

Figura 1	Sobreposição de informações no mapa do site PortoAlegre.cc	11
Figura 2	MapATL não possui sobreposição de informação devido ao filtro por zoom	12
Figura 3	Entre ZOOM A e ZOOM C existem muitos níveis intermediários e não apenas 1 (ZOOM B).	12
Figura 4	Informações necessárias em níveis superiores, de zoom, visualizadas em níveis inferiores	13
Figura 5	O centro do mundo na latitude 0 e longitude 0 reside em algum lugar a oeste da costa da África	16
Figura 6	A cada incremento de zoom um mapa dobra a sua resolução de imagens	17
Figura 7	Busca como estratégia para redução do número de marcadores	19
Figura 8	Filtro como estratégia para redução do número de marcadores	20
Figura 9	Otimização Visual como estratégia para redução do número de marcadores	21
Figura 10	Estes dois marcadores não serão agrupados pois residem em quadrados diferentes da grade	22
Figura 11	Usando tamanho e cor do marcador para representar a dimensão do grupo	24
Figura 12	Usando gráficos para representar agrupamentos	24
Figura 13	Principais algorítimos de agrupamentos de pontos	25
Figura 14	Exemplo de utilização da API MarkerClusterer	29
Figura 15	Página do github.com que hospeda o código fonte deste projeto.	31
Figura 16	Arquitetura do framework Searchlight.	33
Figura 17	Processamento de dados do framework Searchlight.	34

Lista de tabelas

Sumário

1	Introdução	8
1.1	Contextualização	9
1.2	Definição do Problema	10
1.2.1	Sobreposição de Informações	10
1.2.2	Zoom arbitrário	11
1.2.2.1	Informações arbitrárias	12
1.3	Objetivos	13
1.4	Contribuições	13
1.5	Estrutura da Monografia	14
2	Fundamentação Teórica	15
2.1	Fundamentos de Mapas Web	16
2.1.1	Coordenadas Geográficas	16
2.1.2	Zoom	17
2.2	Estratégias para lidar com muitos marcadores	18
2.2.1	Reducindo o número de marcadores	18
2.2.1.1	Busca	18
2.2.1.2	Filtro	19
2.2.1.3	Otimização Visual	20
2.2.2	Agrupamento/Clustering	21
2.2.2.1	Agrupamento Por Grade	22
2.2.2.2	Agrupamento Por Distância	22
2.2.2.3	Agrupamento Por Região	23
2.2.2.4	Estilos de visualização de agrupamento	23
2.3	Algorítimos de agrupamento	25
2.3.1	Métodos baseados em grade	26
2.3.1.1	WaveCluster	26
3	Desenvolvimento	27
3.1	Considerações Iniciais	28
3.1.1	Agrupamento de Pontos	29
3.1.1.1	MarkerClusterer	29
3.1.1.2	Leaflet.MarkerCluster	29
3.2	Ferramentas Utilizadas	30
3.2.1	Github	30
3.2.2	Leaflet.js	31

3.2.3	TableTop.js	32
3.2.4	Rapydscript	32
3.3	Arquitetura do Sistema	32
3.3.1	Processamento de dados	34
4	Solução Desenvolvida	36
4.1	Site do projeto	37
4.2	Recursos desenvolvidos	37
4.2.1	Filtro por categorias	37
4.2.2	Agrupamento de marcadores	37
4.2.3	Foco em Grupo de marcadores	37
4.2.4	Geração automática de mapas	37
4.2.4.1	Marcadores com link,imagens e videos	37
4.2.5	Compartilhamento de mapas	37
5	Conclusão	38
5.1	trabalhos futuros	39
5.1.1	Redes de Grupo	39
5.2	Considerações finais	39
Referências		40
Apêndices		42
Anexos		43

1 Introdução

1.1 Contextualização

O estado atual da comunicação mundial permite as pessoas, de qualquer país, comunicar e trocar conhecimento dos mais diversos assuntos. Essa facilidade de comunicação também facilita a união de indivíduos, que não se conhecem e com realidades sociais completamente opostas, em um projeto com metas em comum.

Um exemplo, dessa união incomum, é o caso de um carpinteiro sul-africano que, ao perder os dedos numa serra, e ficar insatisfeito com os preços e qualidade das próteses disponíveis no mercado, resolveu iniciar um projeto open source de mão biônica com a ajuda de um técnico de efeitos especiais que mora nos EUA. Juntos, esses dois indivíduos que até então não se conheciam, conseguiram criar um projeto de mão biônica por 150 dólares. Após isso, o projeto ganhou visibilidade e patrocínio e já está sendo testado por algumas crianças deficientes da África do Sul¹.

Essa melhoria de comunicação, obviamente, também afeta o campo empresarial. Em algumas empresas, o sistema de "Outsourcing", que é um sistema de aquisição de conhecimento ou tecnologia, começou a ser substituído pelo sistema de "Crowdsourcing", que no campo empresarial, funciona como uma espécie de concorrência: a empresa lança uma espécie de edital informando quanto pode pagar por determinada solução e quem quer que seja poderá oferecer a resposta adequada ao que se procura, independente de ser uma, duas, ou cem pessoas, contanto que resolva o problema.

De modo geral Crowdsourcing é a prática de obtenção de serviços, idéias ou conteúdo solicitando contribuições de um grande grupo de pessoas e, especialmente, a partir de uma comunidade on-line, ao invés de funcionários ou fornecedores tradicionais². Existem diversos tipos de crowdsourcing, mas nesse trabalho iremos nos focar no crowdsourcing conhecido como *Wisdom of the Crowd* que é um tipo de crowdsourcing que coleta grandes quantidades de informação e as agrupa para obter uma visão completa e precisa sobre um determinado tema. Essa visão, dependendo do tema de estudo, pode ser representada por um mapa de crowdsourcing.

A produção de mapas crowdsourcing é geralmente feita de forma automática, usualmente temos algum software e/ou site que coleta as informações e as agrupa por meio de algum algorítimo desenvolvido especificamente para o mapa.

A coleta dessas informações pode ocorrer de várias formas, tanto manual como automática. Por exemplo, no site PortoAlegre.cc os usuários podem adicionar novas informações através do site, ou seja, ela é feita de forma manual. Alguns sistemas podem coletar informações de forma automática usando programas de computadores, aplicativos

¹ Notícia sobre projeto de mão biônica <http://meiobit.com/115807/>

² Definição completa de crowdsourcing <http://en.wikipedia.org/wiki/Crowdsourcing>

de smartphones ([THIAGARAJAN et al., 2010](#)) e até mesmo da internet³.

Uma vez coletada, essa informação é analisada e exibida em um mapa, mas o mapa em si não é o produto final, e sim algumas informações específicas retiradas dele. Por exemplo, podemos ter mapas que mostrem os congestionamentos no trânsito de uma cidade e um sistema ([THIAGARAJAN et al., 2009](#)) que através desse mapa consegue identificar uma rota mais eficiente com menor consumo de energia, evitando assim ficar parado no congestionamento gastando gasolina.

Em alguns casos, mapas de crowdsourcing possuem informações posicionadas em regiões muito próximas entre si, que devido a quantidade elevada, acabam poluindo a visualização e dificultando a compreensão do mapa. Esse problema pode ser resolvido quando os mapas oferecem mecanismos para agrupar e filtrar essas informações. Um mecanismo ideal é o zoom contextual ou zoom em grupo, que filtra informações irrelevantes, em determinados níveis de zoom, deixando o mapa mais leve e comprehensível.

O Projeto Searchlight pretende ser uma ferramenta para auxiliar e melhorar a visualização de mapas de crowdsourcing.

O escopo do projeto atinge a criação de uma ferramenta que visualize mapas de crowdsourcing em um navegador de internet, tanto desktop quanto mobile, usando recursos de visualização de mapas já disponíveis em HTML5 mas que ainda não possuem zoom contextual e outras opções úteis que permitam uma melhor visualização do mapa.

1.2 Definição do Problema

Mapas de crowdsourcing tendem a mostrar uma enorme quantidade de informação. Essa característica faz com que, em alguns casos, a visualização e a compreensão do mapa seja comprometida.

Ao trabalhar com mapas de crowdsourcing, geralmente encontramos 2 problemas: a sobreposição de informações e o zoom arbitrário.

1.2.1 Sobreposição de Informações

O site [PortoAlegre.cc](#) é um exemplo da importância do mapas de crowdsourcing no contexto governamental e na sociedade. Por meio desse site os cidadãos de porto alegre podem relatar os problemas de sua cidade para que as autoridades tomem as devidas providências.

Um dos principais objetivos do site é identificar as áreas prioritárias em que o governo deveria atuar. Mas a sobreposição de informações dificulta essa tarefa, pois

³ Mapa com informações coletadas do Twitter <http://trendsmap.com>

ocorre frequentemente nesse site.

Figura 1 – Sobreposição de informações no mapa do site PortoAlegre.cc



Fonte: PortoAlegre.cc (2013)

Esse problema fica evidente na [Figura 1](#) quando consideramos, a possibilidade, que um grupo de 5 marcadores reunidos numa região específica, podem sobrepor dezenas ou até milhares de outros marcadores. Ou seja, o mapa não consegue mostrar, com clareza e precisão, as áreas de maior ocorrência de determinado incidente.

O site fornece um filtro por categorias, que diminui de forma significativa a quantidade de informação exibida. Mas infelizmente não resolve o problema, pois a sobreposição de informação ainda pode ocorrer com informações de uma mesma categoria.

1.2.2 Zoom arbitrário

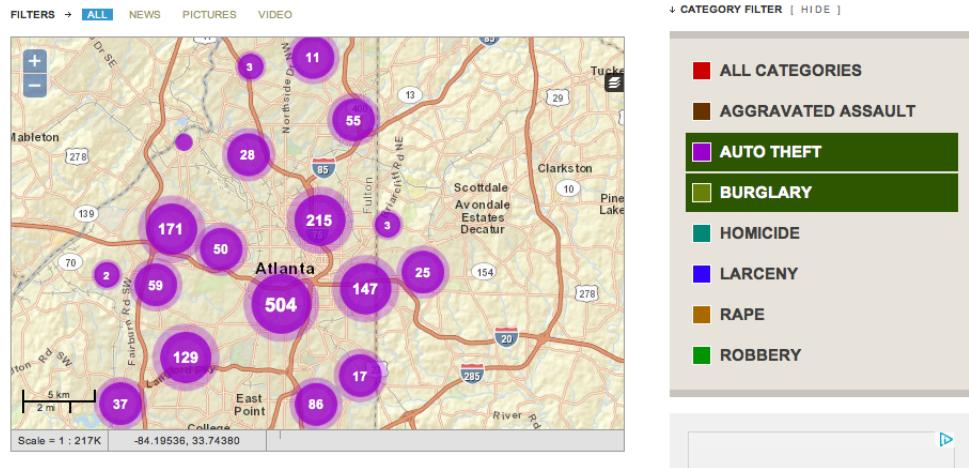
Alguns sites criam mecanismos que minimizam o problema da sobreposição de informações. Como exemplo, temos o [MapATL](#) (2013) mostrado na [Figura 2](#) que mostra um mapa com a taxa de crimes em Atlanta.

O site fornece filtros por categoria, data e zoom. Mas o principal responsável pela eliminação da sobreposição de informação é o filtro por zoom. Esse filtro agrupa todos os marcadores que estão sobrepostos, no zoom atual, em um único marcador que exibe a informação somada dos marcadores que o compõem.

Segundo ([SILVA, 2010](#), 42,44) esse tipo de agrupamento, baseado em grelha, pode ser implementado a partir do algorítimo WaveCluster([SHEIKHOLESLAMI; CHATTERJEE; ZHANG, 2000](#)).

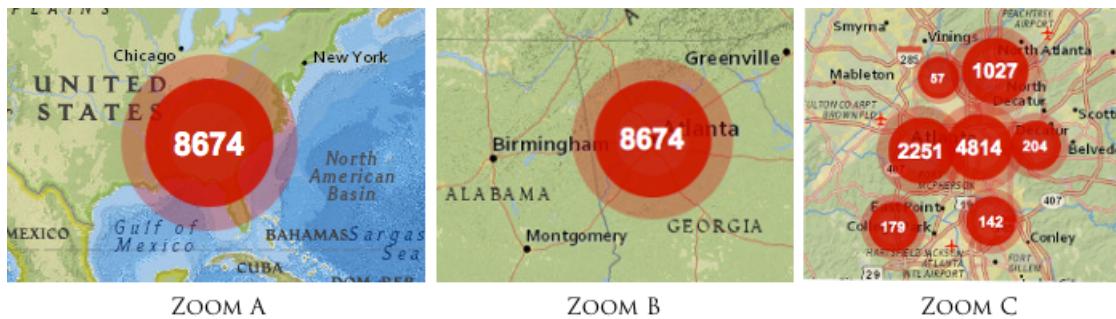
Esse algorítimo resolve o problema de sobreposição de informações. Porém o problema do zoom arbitrário, ilustrado na [Figura 3](#), permanece.

Figura 2 – MapATL não possui sobreposição de informação devido ao filtro por zoom



Fonte: MapATL (2013)

Figura 3 – Entre ZOOM A e ZOOM C existem muitos níveis intermediários e não apenas 1 (ZOOM B).



Fonte: MapATL (2013)

O usuário precisa aplicar vários zoons para ir do ZOOM A para o ZOOM C. Mas durante essa interação é gasto tempo e banda, da conexão de internet, do usuário para exibir os zoons intermediários, quando o ideal seria exibir apenas o ZOOM B.

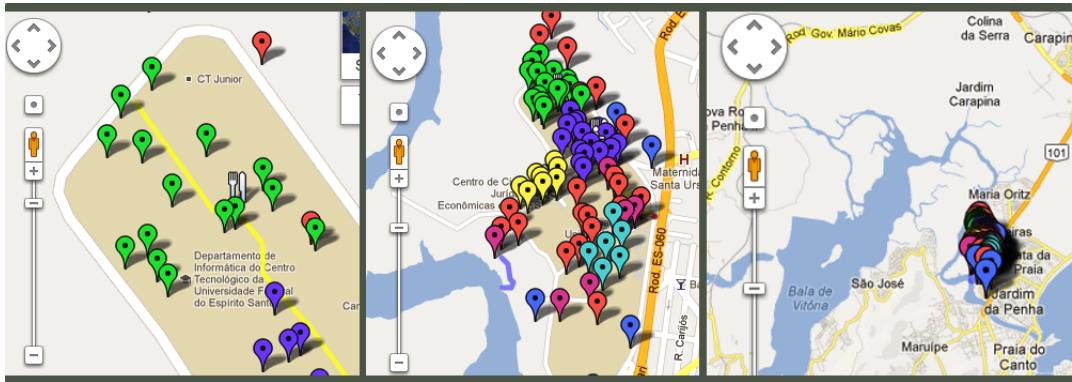
Esse gasto de banda, prejudica a usabilidade de mapas em dispositivos móveis pois, geralmente, eles possuem pouca banda de internet.

Uma abordagem para esse problema é o uso de um zoom inteligente que siga uma hierarquia espacial invés de simplesmente dobrar a visualização atual.

1.2.2.1 Informações arbitrárias

Um subproblema do zoom arbitrário é a exibição de informações desnecessárias em todos os níveis de zoom. Na Figura 4 podemos observar o problema, que neste caso, a arbitrariedade está na exibição de informações, e não no número de zoons. Os marcadores

Figura 4 – Informações necessárias em níveis superiores, de zoom, visualizadas em níveis inferiores



Fonte: <http://maps.google.com>

desse mapa exibem informações sobre os departamentos internos da UFES, mas essas informações são úteis apenas em certos níveis de zoom. Em níveis mais baixos, onde a área dessa universidade seja desprezível, a informação perde seu valor e fica poluindo o mapa. Neste caso, um único marcador representando o grupo seria mais útil.

1.3 Objetivos

Com base nas informações da definição do problema, o objetivo deste trabalho é desenvolver uma ferramenta que: agrupe marcadores de forma inteligente eliminando a sobreposição; forneça um zoom inteligente que não exiba níveis intermediários desnecessários; esconda marcadores quando sua informação não seja mais necessária ao nível observado.

O objetivo principal desse trabalho é facilitar a visualização de informação em mapas crowdsourcing. Isso devido a importância sócio-econômica das informações que geralmente são exibidas nessa classe de mapas. Mas a ferramenta poderia implementar recursos que também facilite a divulgação dessas informações. Por isso, o objetivo secundário é fornecer um mecanismo para gerar mapas a partir de planilhas de dados, e compartilhá-los sem que o autor precise programar ou ter algum conhecimento de programação. Para este trabalho a divulgação dos mapas é um objetivo secundário, mas poderia ser melhor abordado em trabalhos futuros.

1.4 Contribuições

A contribuição deste trabalho é o desenvolvimento de um framework para exibição de mapas em páginas web facilitando a visualização de informações crowdsourcing.

Além disso também foi criado um website do projeto([QUIRINO, 2013](#)) que explica e documenta o framework. O website fornece também um pagina de geração e compartilhamento de mapas por pessoas que não sabem programar.

1.5 Estrutura da Monografia

Este trabalho está organizado da seguinte forma:

Nesta introdução, é apresentado o contexto geral do projeto, a definição do problema, a solução proposta e a estrutura da monografia.

No capítulo 2, Fundamentação Teórica, apresenta-se uma explicação básica sobre os elementos comuns em mapas geográficos, usados na web; uma pesquisa sobre as estratégias para lidar com mapas que possuem muitos marcadores e uma pequena explicação sobre os principais algorítimos para agrupamento de pontos.

No capítulo 3, Desenvolvimento, é explicado como o projeto foi desenvolvido, quais foram as ferramentas utilizadas, e o motivo da escolha de determinadas tecnologias.

No capítulo 4, Solução Desenvolvida, é apresentado em detalhes a solução que foi desenvolvida e como utilizá-la.

No capítulo 5, Conclusão, é apresentado as dificuldades encontradas no decorrer do projeto, trabalhos futuros e conclusão geral sobre o projeto.

2 Fundamentação Teórica

2.1 Fundamentos de Mapas Web

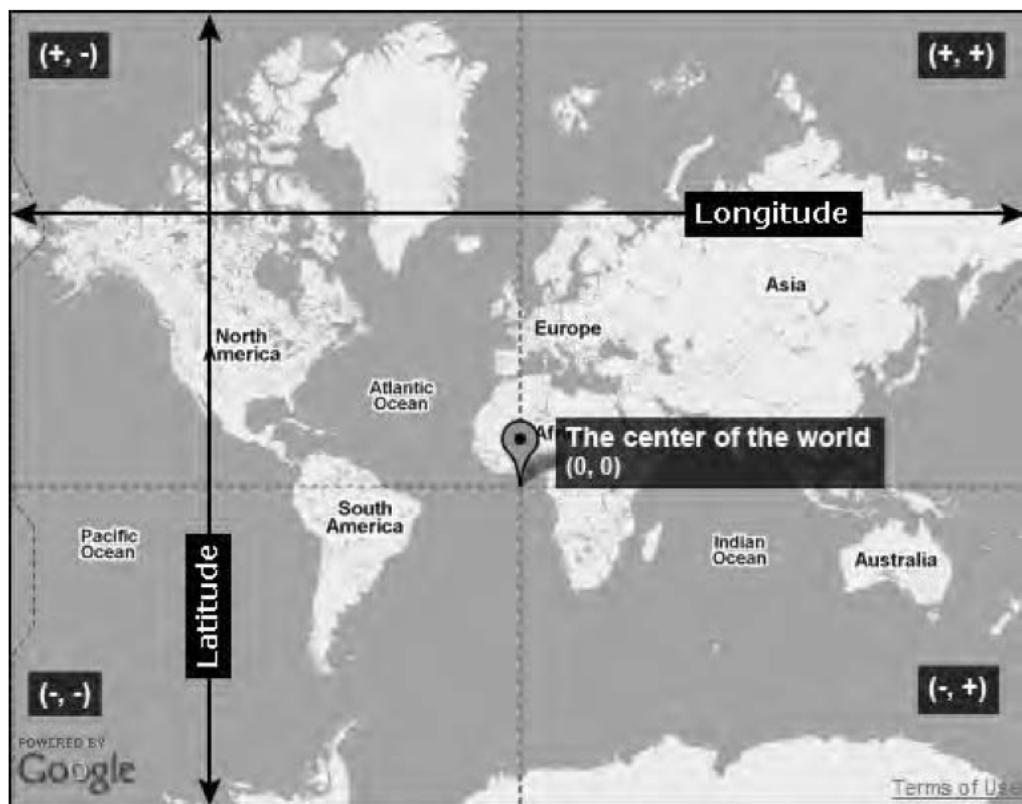
2.1.1 Coordenadas Geográficas

Coordenadas geográficas são usadas para expressar localizações no mundo. Existem vários sistemas de coordenadas diferentes. O sistema de coordenadas usado pelo Google Maps é o Word Geodetic System 84 (WGS84), que é o mesmo sistema que o Global Position System(GPS) usa.

As coordenadas são expressas usando o conceito de latitude e longitude. Onde latitude mede do sul ao norte e longitude mede do oeste para o leste. No equador a latitude é 0. Isso significa que tudo abaixo do equador (hemisfério sul) possui uma latitude negativa, e tudo acima possui uma latitude positiva. Similarmente também existe uma linha zero para longitude. Ela é conhecida como meridiano, e por razões históricas passa por Greenwich, Inglaterra. Cada posição que é localizada a leste desta linha tem um número positivo e tudo a oeste tem um número negativo([SVENNERBERG, 2010, 4](#)).

A [Figura 5](#) permite uma melhor observação desses conceitos:

Figura 5 – O centro do mundo na latitude 0 e longitude 0 reside em algum lugar a oeste da costa da África



Fonte: ([SVENNERBERG, 2010, 5](#))

Dessa forma é possível representar o mapa do mundo em uma imagem retangular

que é projetada sobre o plano cartesiano.

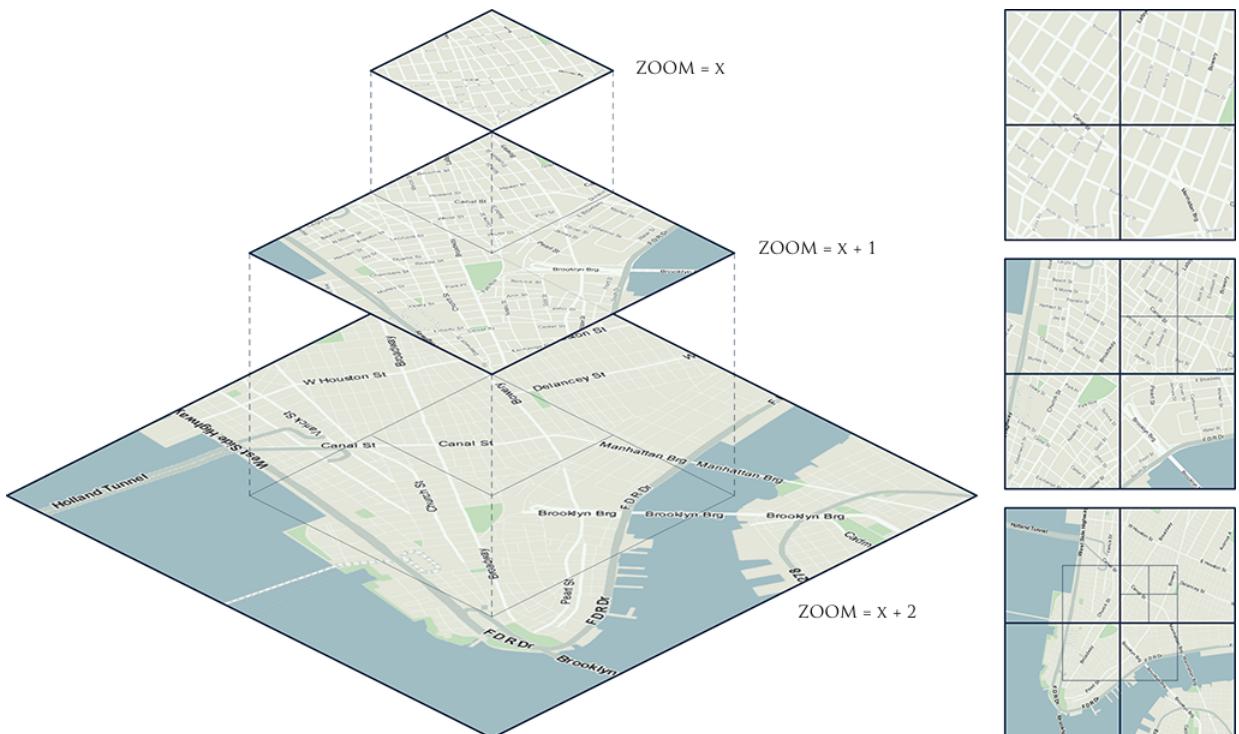
Caso seja necessário um maior grau de detalhamento, do mapa, basta que se aumente a resolução da imagem mantendo as proporções e relações entre as coordenadas (x,y), do plano cartesiano, com as coordenadas (longitude, latitude) que são usadas pelo mapa.

2.1.2 Zoom

As coordenadas de longitude e latitude servem para localizar um ponto numa imagem retangular, através de suas correspondentes x e y do plano cartesiano. Porém mapas também fornecem uma terceira coordenada conhecida como coordenada de Zoom.

Esta coordenada controla qual o tamanho da imagem que será usada para projetar o mapa sobre o plano cartesiano. Usualmente, um mapa com coordenada zoom, ou nível de zoom, igual a zero possui uma imagem de tamanho 256x256 pixels. A cada incremento, da coordenada de zoom, dobra-se o tamanho da imagem usada para representar o mapa, e consequentemente o nível de detalhes. De forma que para zoom=1 a imagem terá 512x512 pixels, para zoom=2 terá 1024x1024 e assim por diante. A [Figura 6](#) exemplifica esse processo:

Figura 6 – A cada incremento de zoom um mapa dobra a sua resolução de imagens



Fonte: <http://workshops.opengeo.org/suiteintro/geowebcache/basics.html>

Portanto para localizar uma posição em um mapa web, de forma detalhada, precisamos de três coordenadas: latitude, longitude e zoom. Além disso, como o tamanho da imagem da projeção varia de acordo com o nível de zoom, o mapa precisa ajustar os valores das coordenadas latitude e longitude para que mantenham suas proporções para os diversos níveis de zoom.

2.2 Estratégias para lidar com muitos marcadores

Um problema comum, ao se trabalhar com mapas de crowdsourcing, é a enorme quantidade de marcadores necessários para representar os dados do mapa. Isto geralmente afeta a performance do mapa, pois quanto mais marcadores são inseridos mais lenta fica a exibição do mapa.

É difícil calcular exatamente o número máximo de marcadores que um mapa suporta antes de começar a ficar lento, pois a velocidade de exibição do mapa depende tanto do navegador quanto do computador em que é exibido. Por exemplo, um mapa pode ser exibido bem rápido em um navegador como o Google Chrome e ao mesmo tempo ser lento quando exibido no Internet Explorer.([SVENNERBERG, 2010, 177](#))

Portanto, é necessário um estudo sobre as estratégias para se lidar com o problema de exibição de muitos marcadores em um mapa. Em ([SVENNERBERG, 2010, capítulo 9](#)) o autor comenta sobre duas estratégias básicas para esse problema. A primeira, e mais óbvia, é reduzir o número de marcadores. A segunda estratégia consiste em agrupar os marcadores por algum grau de similaridades.

2.2.1 Reduzindo o número de marcadores

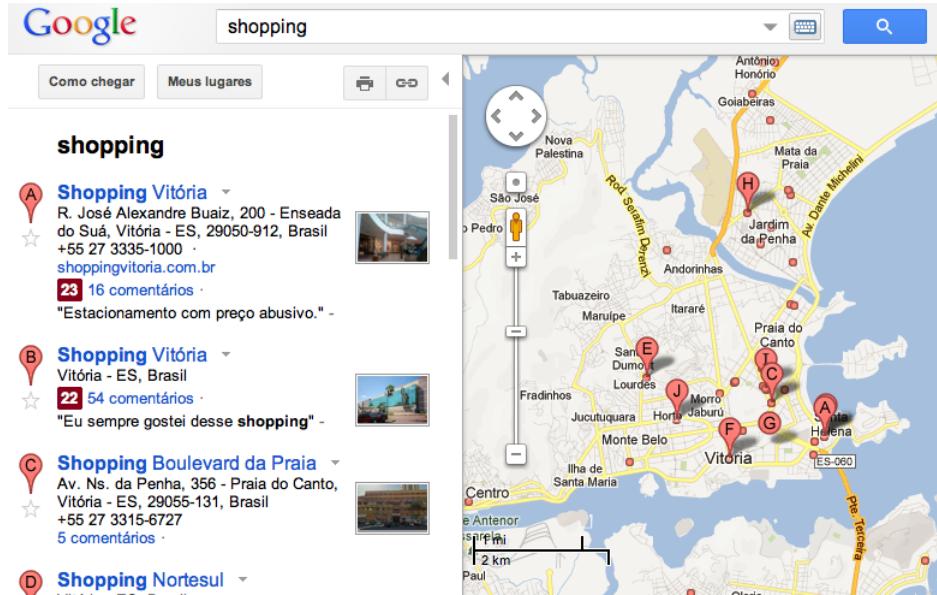
Existem muitos meios de se reduzir o número de marcadores, entre eles destacam-se as reduções via busca, filtro e otimização visual.

2.2.1.1 Busca

Para se reduzir o número de marcadores exibidos no mapa pode-se fornecer um mecanismo de pesquisa ou busca no mapa. Dessa forma mesmo que o mapa possua milhões de marcadores, somente aqueles que satisfaçam os critérios da busca são exibidos.

A [Figura 7](#) exemplifica isso ao mostrar o exemplo de busca no google maps. O google maps possui um catálogo enorme de informações, sobre diversas regiões geográficas. Mostrar todas essas informações ao mesmo tempo deixaria o mapa muito poluído, e praticamente inutilizável. Para que isso não aconteça é implementado o mecanismo de busca.

Figura 7 – Busca como estratégia para redução do número de marcadores



Fonte: <http://maps.google.com>

Na [Figura 7](#) observa-se a exibição de uma busca pela palavra chave shopping. Nesta mesma região de interesse existem estabelecimentos de outras categorias como supermercados, lojas, padarias etc. Mas o google maps exibe apenas os marcadores relativos a categoria shopping, que é a palavra chave da busca. Ao fazer isso o mapa fica mais limpo e compreensivo.

2.2.1.2 Filtro

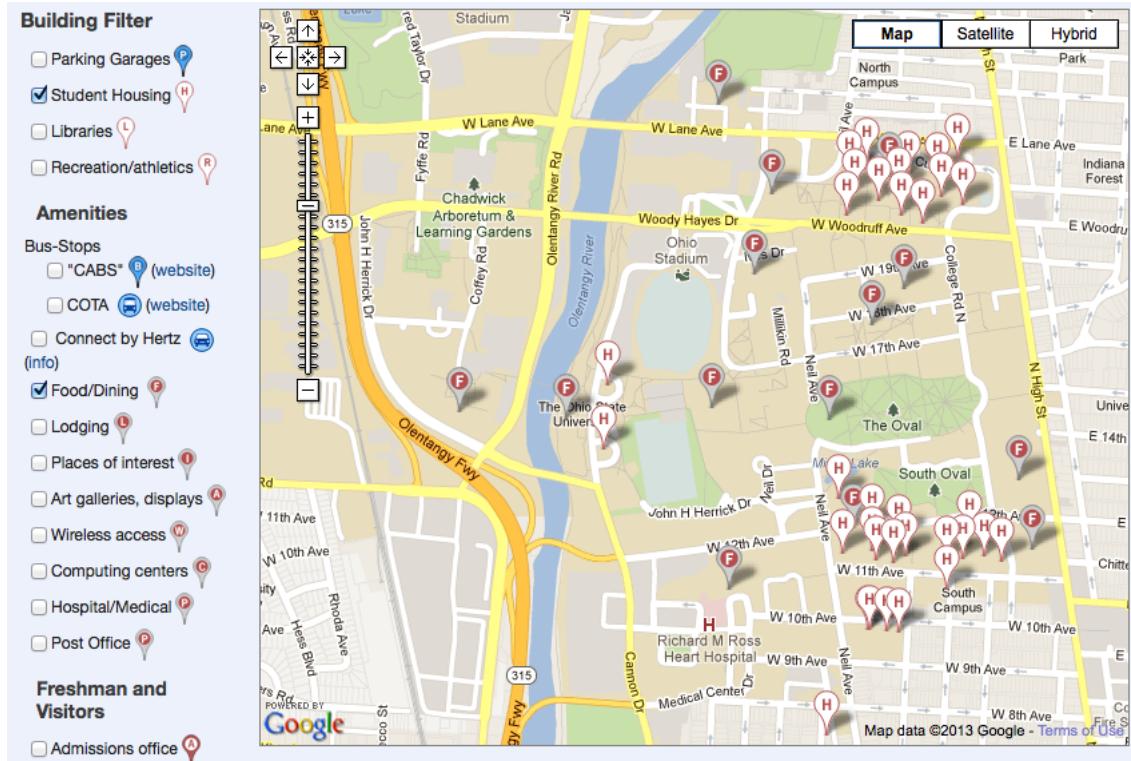
De forma similar ao mecanismo de busca, um mapa também pode possuir um mecanismo para filtrar grupos de marcadores de acordo com algum critério de seleção do grupo. Dessa forma somente os marcadores que pertençam aos grupos marcados no filtro são exibidos no mapa.

Diferente do método de busca, este método permite ser mais específico e direto, pois cada campo do filtro é relacionado diretamente com alguma categoria contida nas informações. No método de busca esta relação é menos direta e depende do algorítimo de busca utilizado.

Como exemplo, ao utilizar-se do método de filtro para marcar duas categorias, obrigatoriamente, deve ser retornado o resultado para as duas categorias que foram marcadas. Já com o método de busca isto nem sempre é verdade, pois dependendo do algorítimo de busca ele pode entender que a busca por “categoria 1 categoria 2” é uma busca por duas categorias ou uma busca por uma categoria conhecida como “categoria 1 categoria 2”, que obviamente não seria encontrada.

A Figura 8 mostra um mapa com filtro a esquerda. Observa-se que apenas os marcadores dos grupos “Student Housing” e “Food/Dining” são exibidos no mapa.

Figura 8 – Filtro como estratégia para redução do número de marcadores



Fonte: <http://www.osu.edu/map/google.php>

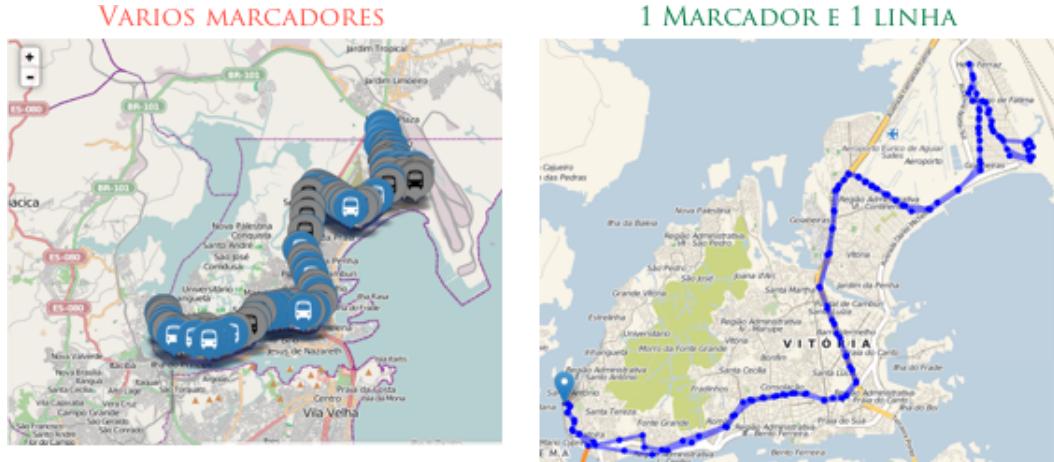
2.2.1.3 Otimização Visual

Nem sempre é preciso usar marcadores para representar dados de um mapa. Em alguns mapas faz mais sentido usar polígonos ou grupo de polígonos para representar um dado ao invés de simplesmente um ponto para o marcador. Por exemplo, ao exibir uma rota não é necessário um marcador para cada vértice da rota, pois precisa-se apenas do desenho de uma linha, passando por cada vértice, para que a rota seja representada.

A Figura 9 mostra um exemplo desse tipo de otimização. No mapa à esquerda, ela exibe uma rota com marcadores em cada vértice do trajeto percorrido. No mapa à direita, ela exibe a mesma rota só que sem colocar marcadores nos vértices do trajeto, deixando apenas um marcador para exibir o ponto de início do trajeto.

Nesse exemplo, cada vértice da rota representa um ponto de parada para os ônibus que percorrem esse trajeto. Inicialmente, pode se achar necessário a exibição de marcadores nesses vértices como mecanismos de interação com os dados que eles representam. Por exemplo, os usuários do transporte público poderiam querer saber quais são os horários

Figura 9 – Otimização Visual como estratégia para redução do número de marcadores



que determinados ônibus passam naquele ponto específico, e com um simples clique no marcador o mapa poderia exibir essa informação.

Porém, para que este tipo de interação seja possível o mapa teria que exibir os marcadores em cada vértice, que como mostrado na [Figura 9](#) não é o melhor caminho. Felizmente, existe algumas alternativas para esse processo que não exigem a presença de um marcador. Por exemplo, ao clicar na linha da rota o mapa pode fazer uma busca, baseada em proximidade, e localizar o ponto que o usuário deseja saber mais informações. Após isso o mapa pode até exibir um marcador temporário como reforço visual do local que está sendo exibida a informação.

2.2.2 Agrupamento/Clustering

A segunda estratégia citada por ([SVENNERBERG, 2010](#), capítulo 9) para reduzir o número de informações exibidas por um mapa, é conhecida como clustering, ou em português, agrupamento.

A estratégia de agrupamento de marcadores consiste em agrupar marcadores que estejam próximos, por algum critério de proximidade, e exibir apenas um marcador para cada grupo e um marcador para cada marcador individual, que não pertença a nenhum grupo. Ou seja, em vez de exibir um marcador individual para cada marcador são exibidos marcadores para grupos e marcadores individuais.

Nessa estratégia os grupos variam de acordo com o zoom. Ou seja, ao fazer zoom em grupo, o grupo se divide em subgrupos menores e, se for o caso, em marcadores individuais.

O processo contrário também ocorre ao diminuirmos o zoom de um grupo. Neste caso o grupo pode se agrupar com grupos vizinhos e formar um novo grupo como mais

elementos, de forma, que em alguns mapas, no zoom=0 o mapa exibe apenas um marcador representando o grupo com todos os marcadores do mapa.

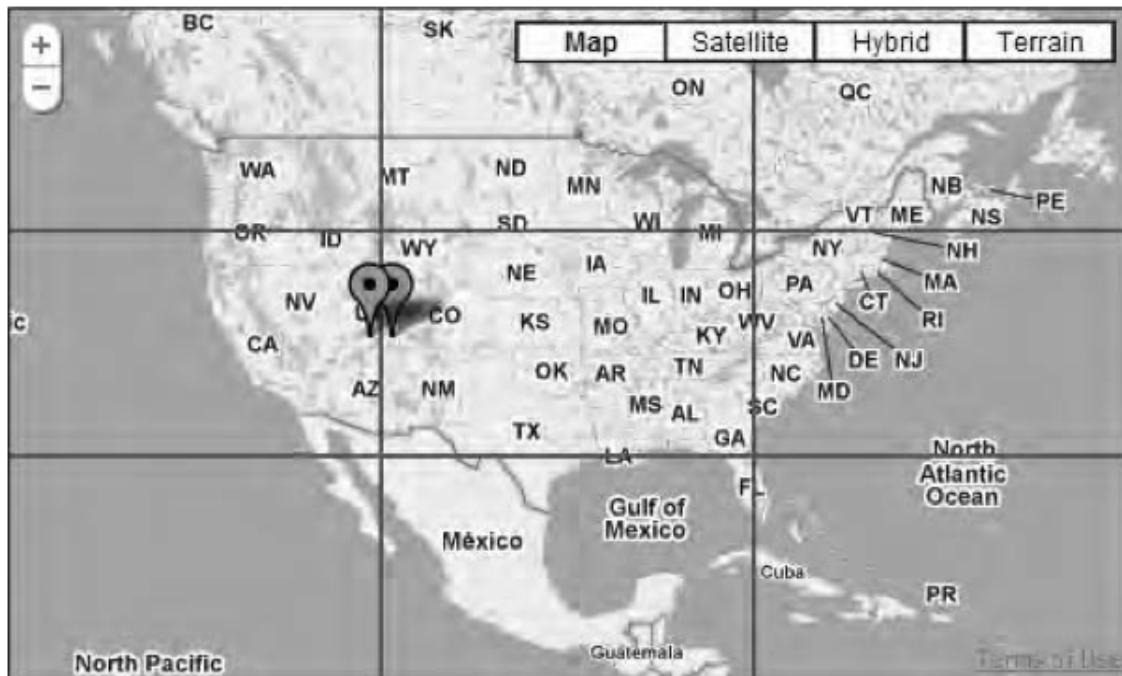
Existem diversos métodos para agrupamento de marcadores, mas os mais comuns são implementados usando algum dos seguintes parâmetros para agrupamento: agrupamento por grade, por distância, ou por região.

2.2.2.1 Agrupamento Por Grade

Agrupamento baseado em grade é provavelmente a abordagem mais comum para agrupar marcadores. Esta abordagem, divide o mapa em uma grade e agrupa todos os marcadores de cada quadrado em um grupo.

Apesar de ser uma técnica eficiente, ela possui limitações óbvias, pois pode levar a resultados indesejados. Por exemplo, considere dois marcadores próximos mas localizados em quadrados diferentes da grade. Neste caso, eles não serão agrupados no mesmo grupo. Para exemplificar isso, veja a [Figura 10](#).

Figura 10 – Estes dois marcadores não serão agrupados pois residem em quadrados diferentes da grade



Fonte: ([SVENNERBERG, 2010](#), figura 9.6)

2.2.2.2 Agrupamento Por Distância

Nesta técnica, não ocorre o problema comentado no agrupamento por grade, pois nela é observado cada marcador e para cada um é procurado marcadores vizinhos, se os marcadores forem próximos o suficiente eles são colocados no mesmo grupo.

Alguns podem achar esta técnica problemática para certos mapas. Pois por agrupar marcadores próximos entre si, os grupos não possuem uma localização fixa como acontece no agrupamento por grade. Assim no agrupamento por distância os grupos podem aparecer em posições aleatórias que podem não fazer sentido para o usuário que observa o mapa.

2.2.2.3 Agrupamento Por Região

No agrupamento por região, define-se diferentes regiões geográficas como países, estados, cidades. E todos os marcadores, de cada região, podem ser agrupados em um grupo para representar a região. Além disso, também é possível definir em que nível de zoom os grupos serão quebrados em subgrupos.

A vantagem é que fica fácil criar grupos que fazem mais sentido para o usuário. Pois, um agrupamento que siga a ordem País > Estado > Cidade é mais natural para o usuário do que um que considere apenas a proximidade.

A desvantagem dessa técnica é o esforço para implementá-la (SVENNERBERG, 2010, 182) já que a definição dos grupos não pode ser facilmente automatizada. Essa dificuldade ocorre devido a natureza de como é organizado a hierarquia das regiões, que varia de país para país, sendo necessário em alguns casos o uso de tabelas para conversão¹.

2.2.2.4 Estilos de visualização de agrupamento

A representação mais simples de um agrupamento pode ser feita por meio de um marcador. Porém, o marcador por si só não expressa muita informação. Não é possível saber a dimensão do grupo ou a natureza de seus elementos observando apenas um marcador.

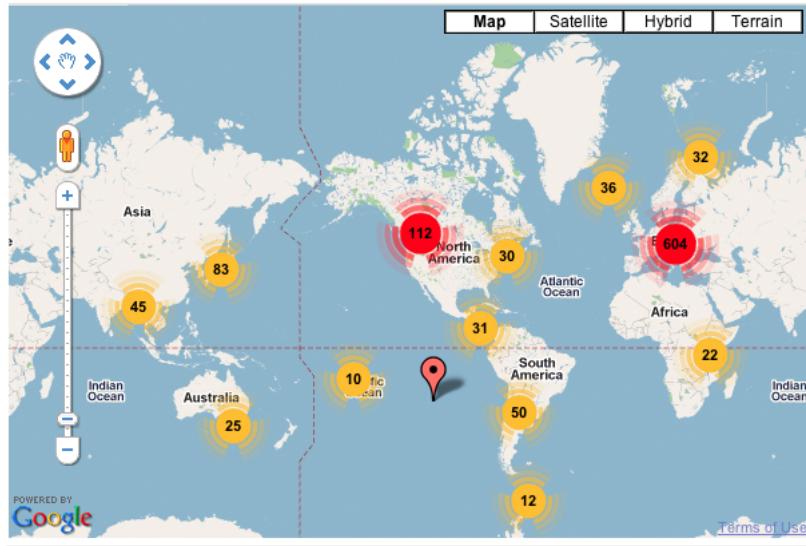
Uma alternativa é usar marcadores com tamanhos diferentes de acordo com o tamanho do grupo. Mas, neste caso, tem-se o problema de ter que classificar um tamanho padrão para determinados marcadores. Pode-se, também, usar cores para representar a dimensão do grupo, onde grupos menores tenderiam para o verde e grupos maiores para o vermelho.

Uma solução bastante comum é uma mistura das soluções anteriores. Onde usa-se um marcador com tamanho diferente de acordo com o número de elementos e cores para maximizar a percepção. Além disso, acrescenta-se uma legenda exibindo o número total de elementos do grupo. Veja a Figura 11

Alguns mapas, possuem dados que permitem uma visualização mais complexa. Para esses tipos de mapa é possível substituir o marcador por uma imagem, gráfico ou

¹Até a definição do que é uma autoestrada precisa de uma tabela de conversão entre países: http://wiki.openstreetmap.org/wiki/Highway:International_equivalence

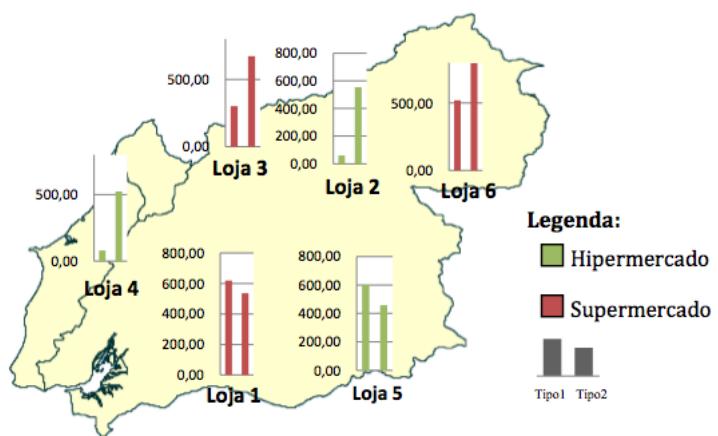
Figura 11 – Usando tamanho e cor do marcador para representar a dimensão do grupo



Fonte: <https://developers.google.com/maps/articles/toomanymarkers>

forma que represente o grupo. Por exemplo, uma mapa que mostre as compras de todos os clientes de uma rede de supermercados poderia agrupar os dados inicialmente por loja e na visualização exibir um gráfico de vendas para representar cada grupo. Veja a Figura 12.

Figura 12 – Usando gráficos para representar agrupamentos



Fonte: (SILVA, 2010, figura 65)

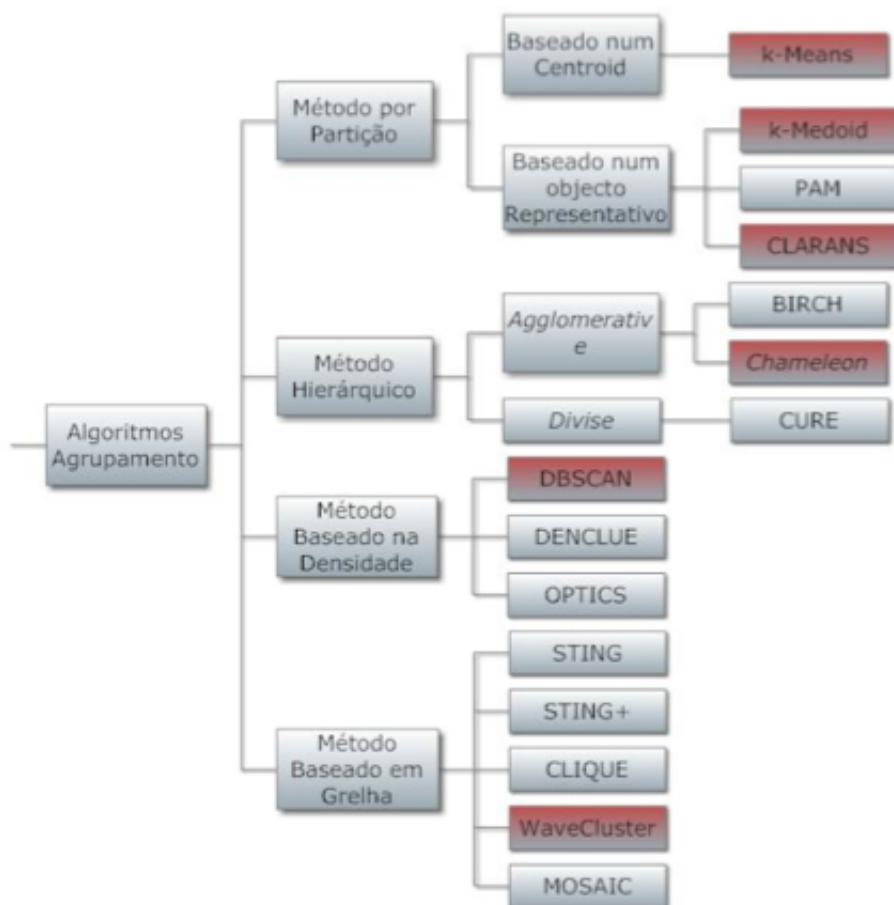
Além disso, os gerentes dessas lojas, poderiam efetuar um zoom no gráfico da loja que gerenciam e ver os subgrupos divididos por bairros, com cada grupo exibindo o gráfico de vendas do bairro, permitindo aos gerentes a criação de uma estratégia de marketing localizada.

2.3 Algorítimos de agrupamento

Basicamente os algorítimos de agrupamentos de pontos podem ser classificados em quatro categorias: (i) método por partição (ii) método hierárquico (iii) método baseado em densidade (iv) método baseado em grade. Isto é ilustrado na [Figura 13](#) onde SILVA destaca os potenciais melhores algorítmos para este tipo de problema. Com exceção dos algorítimos k-Means e k-Medoid que foram introduzidos apenas por motivos históricos ([SILVA, 2010](#), 35).

Observa-se que em ([SILVA, 2010](#), capítulo 2) é feito um estudo mais aprofundado sobre os principais algorítimos de agrupamentos de pontos. Por isso, nesta seção será descrito apenas o algorítimo que engloba o contexto deste trabalho. Ou seja, o algorítimo WaveCluster que é baseado no método de grade.

Figura 13 – Principais algorítimos de agrupamentos de pontos



Fonte: ([SILVA, 2010](#), figura 15)

2.3.1 Métodos baseados em grade

Como já foi comentado na [Subseção 2.2.2](#), agrupamentos baseados em grade são aqueles que dividem o espaço do mapa em uma grade e agrupa os marcadores de cada quadrado ou célula da grade. Ou seja, algorítimos baseados em grade são aqueles que usam o agrupamento por grade como parâmetro de agrupamento. Pode se observar na [Figura 13](#) que os principais algorítimos são: STING, STING+, CLIQUE, WaveCluster e MOSAIC. E, apesar de não ser o mais indicado para o contexto SOLAP+ que SILVA estudava, o melhor entre os algorítimos baseados em grade é o WaveCluster.

2.3.1.1 WaveCluster

Uma boa abordagem para agrupamentos deve ser eficiente e detectar grupos de formas arbitrárias. Ela também tem que ser insensível a dados discrepantes e a ordem de entrada dos dados.

O algorítmico WaveCluster atende todos esses requisitos. Sua definição completa é encontrada em ([SHEIKHOLESLAMI; CHATTERJEE; ZHANG, 2000](#)) mas em resumo o algorítmico faz o seguinte:

Entrada Vetores de características dos objetos de dados multidimensionais
Saida Objetos agrupados

1. Quantizar o espaço de características, atribuir objetos as celulas da grade.
2. Aplicar a transformação *wavelet*¹ no espaço de características quantizado.
3. Encontrar os componentes conectados (clusters) nas sub-bandas
4. Transformar o espaço original, em diferentes níveis.
5. Atribuir rótulos as células.
6. Criar a tabela de pesquisa.
7. Mapear objetos aos clusters/grupos.

Algorithm 1: WaveCluster

O WaveCluster possui complexidade $O(n)$. Além disso, por causa do uso das técnicas de processamento de sinais a propriedade de multi-resolução, ou agrupamento nos níveis hierárquicos, também é aplicada ao WaveCluster.

O seu uso não é limitado apenas à dados espaciais, o algorítmico WaveCluster pode ser aplicado a qualquer conjunto de atributos com valores numéricos ordenados.

¹Uma transformada de wavelet é uma tecnica de processamento de sinais que decompõe o sinal em diferentes sub-bandas de frequencia.

3 Desenvolvimento

Neste capítulo será detalhado o processo de desenvolvimento do framework Searchlight, e as ferramentas utilizadas para a construção do mesmo.

3.1 Considerações Iniciais

Durante a fase de pesquisa deste projeto, descobriu-se uma variedade de recursos que poderiam ser usados na elaboração de um framework para visualização de mapas. Muitos destes recursos já estavam no cronograma do projeto, outros foram adicionados e alguns ficaram fora do cronograma e colocados na lista de trabalhos futuros.

Inicialmente a meta do framework era apenas encontrar uma maneira de visualizar mapas de crowdsourcing de forma mais limpa e sucinta através de técnicas de agrupamento aliadas a hierarquia de zoom. Mas no decorrer das pesquisas percebeu-se que seria interessante adicionar alguns recursos como filtro por categoria e foco em grupo.

Também foi adicionado ao projeto a possibilidade de gerar e compartilhar um mapa web sem precisar escrever uma linha de código. O único requisito é fornecer um endereço web onde o gerador de mapas deve buscar os dados geográficos do mapa. Neste caso, o usuário não precisa mais saber programar para poder usufruir dos recursos do framework e a única preocupação dele fica depositada sobre o conteúdo do mapa.

Em relação ao conteúdo usado pelo compartilhador de mapas, o framework atualmente suporta o armazenamento numa planilha eletrônica ou em um arquivo de formato JSON¹. Quando o conteúdo é oriundo de planilha eletrônica é necessário que esta esteja armazenada no Google Docs e seja pública. De forma similar, quando o conteúdo é armazenado em um arquivo JSON é necessário que o website que hospeda o arquivo suporte o protocolo JSONP. Caso contrário, o usuário deve instalar o framework Searchlight no website que deseja visualizar o mapa.

Em relação a API de mapas utilizada pelo projeto, o objetivo inicial era usar a api do Google Maps para a visualização dos mapas. Porém, na época em que esta pesquisa era feita, circulava notícias que o Google estaria mudando seus termos de serviço e iria começar a cobrar² pelo uso da API do Google Maps. Além disso, no mesmo período, a Apple anunciou que iria abandonar o uso do Google Maps em seus smartphones e adotaria uma solução própria. Isto trouxe a necessidade de procurar uma alternativa ao Google Maps. A alternativa encontrada foi a biblioteca open-source Leaflet.js (LEAFLET, 2013) que será descrita no decorrer deste capítulo.

¹Formato muito usado para troca de dados entre websites <http://www.json.org>

²Uma notícia sobre o inicio das cobranças do Google Maps <http://goo.gl/f1E3K>

3.1.1 Agrupamento de Pontos

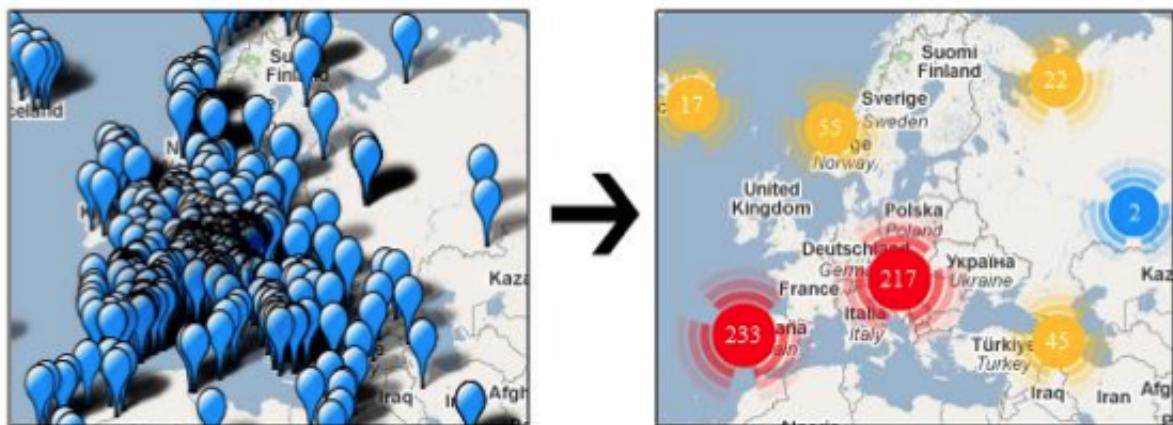
Durante a fase de pesquisa, deste trabalho, foi feito uma busca por ferramentas acadêmicas, e comerciais, que já fizessem o uso de algorítimos de agrupamento de pontos. Foram encontradas duas ferramentas promissoras: MarkerClusterer e Leaflet.MarkerCluster.

3.1.1.1 MarkerClusterer

MarkerClusterer ([SVENNERBERG, 2010](#), 188) é uma ferramenta fornecida pelo google que estende o uso da API do Google Maps para o uso de algorítimos de agrupamentos.

MarkerClusterer é uma solução baseada em grelha. Ela agrupa marcadores de acordo com sua distância ao centro de um grupo. Quando um marcador é adicionado, ele pesquisa sua posição em todos os grupos. Caso não seja colocado em nenhum grupo, um novo grupo é criado para este marcador. A [Figura 14](#) demonstra a aplicação da ferramenta nos marcadores de um mapa.

Figura 14 – Exemplo de utilização da API MarkerClusterer



Fonte: ([SILVA, 2010](#), figura 20)

3.1.1.2 Leaflet.MarkerCluster

Assim como a biblioteca MarkerCLusterer estende a API do Google Maps para uso de agrupamento de marcadores, o plugin Leaflet.MarkerCluster([LEAFLET.MARKERCLUSTER, 2013](#)) serve como uma extensão para a API Leaflet.

Este plugin, faz basicamente as mesmas funções da ferramenta MarkerCluster, porém com algumas melhorias. Por exemplo, ao fazer zoom o mapa exibe uma pequena animação do processo de agrupamento; ao se passar o mouse sobre um marcador do grupo

o mapa exibe um polígono que mostra os limites alcançados pelo grupo; os grupos que não são visíveis na visualização atual do mapa são retirados do mapa para aumentar a performance. Além dessas melhorias, uma que merece destaque é a sua incrível capacidade de customização.

Ao contrário da biblioteca MarkerClusterer, o plugin Leaflet.MarkerCluster possui um documentação bem completa que pode ser encontrada em ([LEAFLET.MARKERCLUSTER, 2013](#)).

É importante observar, que inicialmente, procurou-se por uma documentação mais abrangente da biblioteca MarkerCluster do Google Maps, mas até o momento da confecção desse trabalho não foi possível encontrar nada além de alguns poucos exemplos de uso. Isto, e outros fatores, contribuíram para a escolha da biblioteca Leaflet.js como API de visualização de mapas do framework Searchlight.

3.2 Ferramentas Utilizadas

Um framework, em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao contrário das bibliotecas, é o framework quem dita o fluxo de controle da aplicação, chamado de Inversão de Controle³.

De modo geral, um framework é conjunto de ferramentas que trabalham em conjunto. O que une as ferramentas são as regras do framework que tem como objetivo prover as funcionalidades que motivaram a criação do framework.

O framework Searchlight tem como objetivo facilitar a criação e a visualização de mapas de crowdsourcing por meio de automatização do processo de criação do mapa. O público alvo abrange programadores e usuários sem nenhum conhecimento de programação. Para atingir este objetivo o framework faz uso do seguinte conjunto de ferramentas: GitHub, Leaflet.js, Tabletop.js e RapydScript.

3.2.1 Github

GitHub é um Serviço de Web Hosting Compartilhado para projetos que usam o controle de versionamento Git. Mais do que isso, GitHub incorpora recursos sociais e de pesquisa que na prática o transformam em uma rede social para programadores. GitHub incorpora elementos tanto do facebook como do twitter. Por exemplo é possível favoritar um projeto, seguir um programador, ver quais são as tendências de programação, quais projetos estão se destacando, quais projetos se relacionam entre si e muitos outros recursos.

³Definição completa em: <http://pt.wikipedia.org/wiki/Framework>

Neste projeto o github foi usado para hospedar o código fonte da solução desenvolvida e o código latex desta monografia. Todos esses dados podem ser acessados através do endereço <https://github.com/wancharle/Searchlight>.

Na Figura 15 podemos observar a pagina do código fonte do projeto. A partir dela um programador pode reportar um bug, solicitar permissões para contribuir com o projeto, copiar o código fonte e muitos outros recursos.

Figura 15 – Pagina do github.com que hospeda o código fonte deste projeto.

The screenshot shows a GitHub repository page for 'Searchlight'. At the top, there are tabs for 'Code', 'Network', 'Pull Requests (0)', 'Issues (0)', 'Wiki', 'Graphs', and 'Settings'. Below the tabs, a summary message reads: 'Trabalho de Conclusão de Curso apresentado ao Departamento de Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação. — [Read more](#)'. Underneath this, there are buttons for 'Clone in Mac', 'ZIP', 'HTTP', 'SSH', and a text field with the URL 'git@github.com:wancharle/Searchlight.git'. To the right of the URL is a 'Read+Write access' button. Below these controls, there are dropdown menus for 'branch: master', 'Files', 'Commits', 'Branches (2)', and 'Tags'. The main content area is titled 'Searchlight / +' and shows a list of 52 commits. The commits are listed in reverse chronological order, starting with 'correções cap 2' by 'wancharle' 27 minutes ago. Other commits include 'monografia' (27 minutes ago), 'site' (8 days ago), 'site_src' (8 days ago), '.gitignore' (2 months ago), '.merge-ghpages' (2 months ago), and 'README.md' (2 months ago). Each commit entry includes the file name, time, and a brief description.

Além de hospedar os códigos fontes do projeto, o GitHub também participa ativamente do framework Searchlight. Seu papel no framework é realizado através da ferramenta GitHub Pages. Essa ferramenta estende o uso do GitHub para que ele se comporte como um servidor de páginas web simples. Isto permite ao framework Searchlight hospedar a página de geração e compartilhamento de mapas sem que seja preciso contratar um servidor web apenas para esse propósito.

3.2.2 Leaflet.js

Leaflet é uma biblioteca open-source feita em JavaScript para visualização de mapas interativos. Seu design é baseado em simplicidade, performance e usabilidade. Devido a isso é bastante amigável a dispositivos móveis como tablets e smartphones.

Leaflet é uma biblioteca moderna e robusta, que tira vantagem dos recursos de navegadores modernos como o uso de HTML5 e CSS3 mas ainda mantém suporte aos

navegadores antigos. É bastante confiável e sua estabilidade já foi testada por grandes sites da internet como Foursquare⁴ e Flickr.

Neste projeto, a biblioteca Leaflet é utilizada para fornecer os recursos de exibição e interação com mapas na web. Além disso, o framework a extensão **Leaflet.Spin**, para controlar o processo de carregamento de dados, e Leaflet.MarkerCluster, para agrupar marcadores.

3.2.3 TableTop.js

Tabletop([SOMA, 2013](#)) é uma biblioteca JavaScript que se comunica com o Google Docs e permite a leitura de planilhas eletrônicas por websites via Javascript.

O framework Searchlight utiliza a biblioteca Tabletop na geração e compartilhamento de mapas. A função da biblioteca neste caso é ler as planilhas com as informações do mapa e converter os dados para o formato JSON que é um dos formatos de dados mais utilizado em JavaScript.

3.2.4 Rapydscript

Rapydscript ([TSEPKOV, 2013](#)) é um pré-compilador para JavaScript com sintaxe bem próxima da linguagem Python. Sua principal vantagem, é a facilidade de se programar orientado a objetos de forma bem similar como é feito Python.

A maior parte do framework Searchlight é escrita usando a sintaxe do Rapydscript, as partes restantes são escritas em JavaScript nativo.

Decidiu-se usar um pré-compilador para JavaScript pois a linguagem JavaScript, apesar de ser uma linguagem com suporte a orientação a objetos, não possui uma sintaxe agradável para criação de classes de objetos. Devido a isso, é possível encontrar na internet algumas dezenas de pré-compiladores de javascript⁵.

3.3 Arquitetura do Sistema

A [Figura 16](#) mostra as camadas que fazem parte da arquitetura do framework Searchlight. A arquitetura do sistema, tal como mostrada na figura, é assim explicada:

- **Camada de Aplicação:** é a camada que disponibiliza os serviços do framework aos usuários. Sendo que para usuários programadores a camada disponibiliza a biblioteca Searchlight.js que permite a integração customizada do framework ao website privado do usuário. Porém, para usuários sem conhecimentos de programação, a

⁴Foursquare abandona Google Maps em prol de leaflet.js <http://goo.gl/JXPCW>

⁵O site <http://altjs.org> exibe uma lista dos principais pré-compiladores de JavaScript.

Figura 16 – Arquitetura do framework Searchlight.



camada também disponibiliza o gerador de mapas. O gerador de mapas, permite a geração automática de mapas e o seu compartilhamento através da internet sem que seja preciso programar.

- **Camada Framework:** é a camada que implementa os serviços oferecidos para os usuários através da camada de aplicação. A camada framework implementa as classes Controle, ClusterCtr e Dados. Essas classes são responsáveis por controlar a interface de usuário; controlar os agrupamentos, filtros e foco; e pelo processamento de dados do framework.
- **Camada Bibliotecas:** é a camada que implementa serviços básicos que são utilizados pela camada framework como visualização de mapas, agrupamentos de marcadores e processamento de dados . Os seus principais componentes são a biblioteca

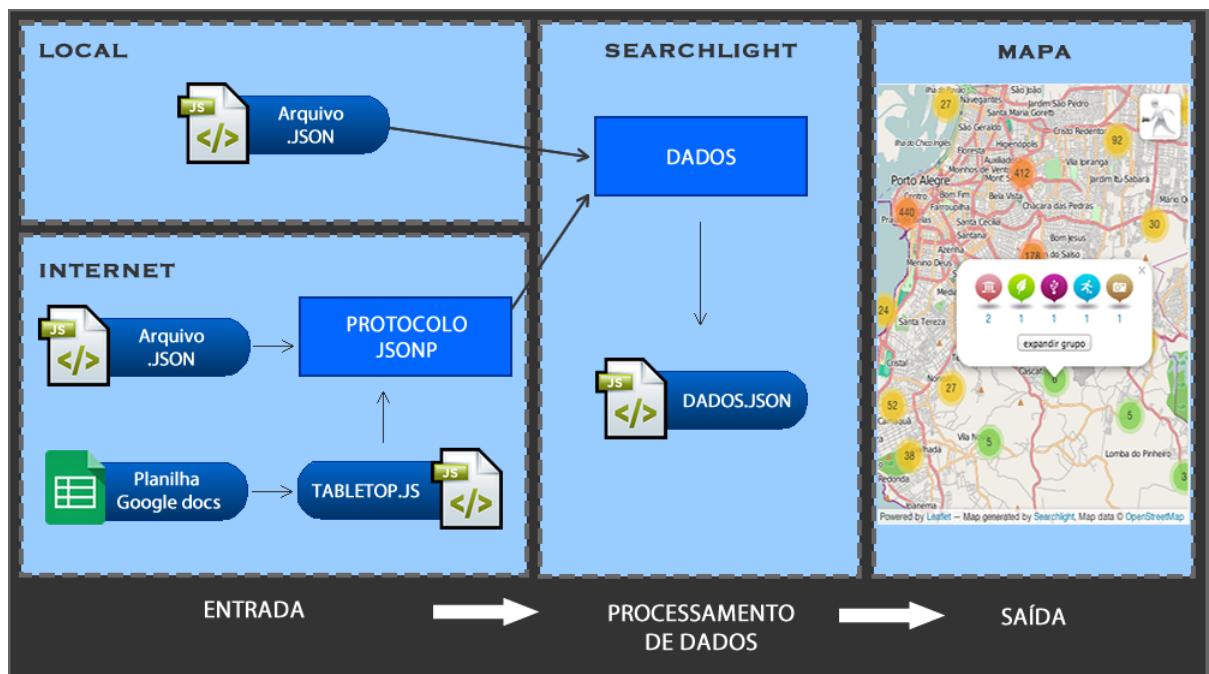
Leaflet, com suas extensões, e a biblioteca Tabletop. Sendo que a biblioteca Leaflet é responsável pelas funções visuais do mapa e a biblioteca Tabletop pelo processamento de dados de planilhas do Google Docs.

- **Camada Base:** é camada que fornece os componentes estruturais para a construção das camadas superiores. Sendo que os seus principais componentes são o jQuery, para fornecer compatibilidade com diversos navegadores, e o grupo HTML, CSS e JavaScript que são os blocos estruturais de todo o framework.

3.3.1 Processamento de dados

Por convenção, o framework Searchlight utiliza o formato JSON como fonte de dados. A entrada dos dados pode ser feia localmente ou pela internet. Dados que não estejam no formato JSON devem ser processados conforme é mostrado na [Figura 17](#).

Figura 17 – Processamento de dados do framework Searchlight.



Vale ressaltar, que os dados fornecidos localmente são utilizados diretamente pelo framework. Porém, os dados que são fornecidos via internet precisam ser processados e seguir as regras do protocolo JSONP.

Arquivos de dados que não estejam no formato JSON podem ser utilizados desde que sejam convertidos e entregues via protocolo JSONP. O framework Searchlight faz essa conversão automaticamente para planilhas do Google Docs. Neste caso, é utilizado a biblioteca Tabletop para converter os dados da planilha para o formato JSON.

S

4 Solução Desenvolvida

4.1 Site do projeto

4.2 Recursos desenvolvidos

4.2.1 Filtro por categorias

4.2.2 Agrupamento de marcadores

4.2.3 Foco em Grupo de marcadores

4.2.4 Geração automática de mapas

4.2.4.1 Marcadores com link,imagens e videos

4.2.5 Compartilhamento de mapas

5 Conclusão

5.1 trabalhos futuros

5.1.1 Redes de Grupo

5.2 Considerações finais

Referências

LEAFLET. *An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps*. maio 2013. [Http://leafletjs.com](http://leafletjs.com). Disponível em: <<http://leafletjs.com>>. Citado na página 28.

LEAFLET.MARKERCLUSTER. *Provides Beautiful Animated Marker Clustering functionality for Leaflet, a JS library for interactive maps*. maio 2013. [Http://github.com/Leaflet/Leaflet.markercluster](http://github.com/Leaflet/Leaflet.markercluster). Disponível em: <<http://github.com/Leaflet/Leaflet.markercluster>>. Citado 2 vezes nas páginas 29 e 30.

MAPATL. *MapATL - Visualizing Atlanta's Crime*. mar. 2013. [Http://crime.mapatl.com/](http://crime.mapatl.com/). Disponível em: <<http://crime.mapatl.com/>>. Citado 2 vezes nas páginas 11 e 12.

PORTOALEGRE.CC. *PortoAlegre.cc é um espaço de colaboração cidadã, onde você pode conhecer, debater, inspirar e transformar a própria cidade. Participe*. mar. 2013. [Http://www.portoalegre.cc/](http://www.portoalegre.cc/). Disponível em: <<http://portoalegre.cc/>>. Citado 3 vezes nas páginas 9, 10 e 11.

QUIRINO, W. S. *Searchlight - Facilitando a visualização de informações em mapas crowdsourcing*. maio 2013. [Http://wancharle.github.com/Searchlight/](http://wancharle.github.com/Searchlight/). Disponível em: <<http://wancharle.github.com/Searchlight/>>. Citado na página 14.

SHEIKHOLESLAMI, G.; CHATTERJEE, S.; ZHANG, A. Wavecluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal—The International Journal on Very Large Data Bases*, 2000. Springer-Verlag New York, Inc., v. 8, n. 3-4, p. 289–304, 2000. Citado 2 vezes nas páginas 11 e 26.

SILVA, R. F. Solap+. 2010. Faculdade de Ciências e Tecnologia, 2010. Citado 4 vezes nas páginas 11, 24, 25 e 29.

SOMA, J. *Tabletop.js takes a Google Spreadsheet and makes it easily accessible through JavaScript*. maio 2013. [Https://github.com/jsoma/tabletop](https://github.com/jsoma/tabletop). Disponível em: <<https://github.com/jsoma/tabletop>>. Citado na página 32.

SVENNERBERG, G. *Beginning Google Maps API 3*. [S.l.]: Apress, 2010. Citado 6 vezes nas páginas 16, 18, 21, 22, 23 e 29.

THIAGARAJAN, A. et al. Cooperative transit tracking using smart-phones. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2010. (SenSys '10), p. 85–98. ISBN 978-1-4503-0344-6. Disponível em: <<http://doi.acm.org/10.1145/1869983.1869993>>. Citado na página 10.

THIAGARAJAN, A. et al. VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. [s.n.], 2009. p. 85–98. Disponível em: <<http://doi.acm.org/10.1145/1644038.1644048>>. Citado na página 10.

TSEPKOV, A. *RapydScript is a pre-compiler for JavaScript, similar to CoffeeScript, but with cleaner, more readable syntax.* maio 2013. [Https://bitbucket.org/pyjeon/rapydscript](https://bitbucket.org/pyjeon/rapydscript). Disponível em: <<https://bitbucket.org/pyjeon/rapydscript>>. Citado na página 32.

Apêndices

Anexos