



**UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
TECNÓLOGO EM REDES DE COMPUTADORES**

**CLYCIA NAJARA SILVA SOUSA**

**IDENTIFICAÇÃO DE DESTINOS TURÍSTICOS BASEADA EM  
DENSIDADE ESPACIAL DE FOTOS**

**QUIXADÁ  
2016**

**CLYCIA NAJARA SILVA SOUSA**

**IDENTIFICAÇÃO DE DESTINOS TURÍSTICOS BASEADA EM  
DENSIDADE ESPACIAL DE FOTOS**

Monografia apresentada ao Curso de  
Redes de Computadores da Universidade  
Federal do Ceará como requisito parcial  
para obtenção do Título de Tecnólogo em  
Redes de Computadores.

Orientador Prof. Msc. Régis Pires Magalhães

**QUIXADÁ  
2016**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

---

S696i Sousa, Clycia Najara Silva  
Identificação de destinos turísticos baseada em densidade espacial de fotos/ Clycia Najara Silva  
Sousa. – 2016.  
37 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de  
Tecnologia em Redes de Computadores, Quixadá, 2016.

Orientação: Prof. Msc. Regis Pires Magalhães

Área de concentração: Computação

1. Cluster (Sistemas de computador) 2. Análise por agrupamento 3. Redes sociais on-line 4.  
Fotografias I. Título.

**CLYCIA NAJARA SILVA SOUSA**

**IDENTIFICAÇÃO DE DESTINOS TURÍSTICOS BASEADA EM  
DENSIDADE ESPACIAL DE FOTOS**

Monografia apresentada ao Curso de  
Tecnólogo em Redes de Computadores da  
Universidade Federal do Ceará como  
requisito parcial para obtenção do grau de  
Tecnólogo. Área de concentração:  
computação

Aprovado em: 11 / Fevereiro / 2016.

**BANCA EXAMINADORA**

---

Prof. MSc. Regis Pires Magalhães (Orientador)  
Universidade Federal do Ceará-UFC

---

Profª. MSc. Ticiane Linhares Coelho da Silva  
Universidade Federal do Ceará-UFC

---

Profª. MSc. Lívia Almada Cruz  
Universidade Federal do Ceará-UFC

Aos meus pais e irmã, por me ajudarem  
em toda a jornada com muita paciência.

Ao meu namorado, pela paciência, ajuda  
e carinho.

## **AGRADECIMENTOS**

Agradeço aos meus pais, por sempre me apoiar e ajudar em toda a jornada, sem eles não teria conseguido chegar até onde cheguei. A paciência e o apoio deles foi fundamental para que eu obtivesse o desempenho desejado.

Aos meus professores por me guiar em todo o ciclo acadêmico e fazer com que eu criasse modelos de comportamento e admiração que irei levar por toda a minha vida.

Aos amigos que ganhei nessa jornada, que me ajudaram, apoiaram, sofreram e torceram em conjunto e que com certeza sempre poderei contar.

Agradeço ao meu orientador, Régis Pires Magalhães pelos conselhos, correções e ajuda imprescindível no trabalho.

"Até mais e Obrigada pelos Peixes."  
(Douglas Adams)

## RESUMO

Este trabalho propõe a clusterização de dados para descobrir pontos de interesse baseados nas fotos do *Flickr*, que pudessem caracterizar possíveis destinos turísticos. Os metadados foram retirados do site, filtrados e transformados em pontos X,Y do plano cartesiano para a realização do experimento. Após isso, foram utilizadas amostras dos clusters e validadas manualmente através do *Google Maps*. O objetivo desse estudo é descobrir possíveis pontos turísticos visitados por usuários utilizando informações armazenadas nos metadados das fotos publicadas.

**Palavras-chave:** Clusterização. Pontos de Interesse. Densidade espacial.

## **ABSTRACT**

This paper proposes a clustering of data to discover points of interest based on photos from Flickr, which could characterize possible tourist destinations. The metadata site were removed, filtered and processed at points X, Y Cartesian plane for the experiment. After that, the clusters samples were used and validated manually by Google Maps. The aim of this study is to discover possible sights visited by users using information stored in the metadata of the published photos.

**Keywords:** Clustering. Points of interest. Spatial Distribution.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de clusterização.....	13
Figura 2 – Funcionamento do DBSCAN.....	15
Figura 3 – Gerador de Bounding Boxes.....	17
Figura 4 – Página da API Explorer do <i>Flickr</i> .....	21
Figura 5 – Exemplo de Arquivo XML retirado do <i>Flickr</i> .....	22
Figura 6 – Representação de clusters no Rio de Janeiro.....	26
Figura 7 – Representação de clusters em Petrópolis.....	26
Figura 8 – Cluster 6. Parque Florestal da Tijuca.....	27
Figura 9 – Cluster 5. Estádio Maracanã.....	28
Figura 10 – Cluster 28. Praia da Barra.....	28
Figura 11 – Cluster 37. Jardim Botânico.....	29
Figura 12 – Cluster 42. Cristo Redentor.....	29
Figura 13 – Cluster1. Museu Imperial.....	30
Figura 14 – Cluster 2. Catedral São Pedro de Alcântara.....	30
Figura 15 – Cluster 8. Palácio Quitandinha.....	31
Figura 16 – Cluster 11. Praça da Liberdade.....	31
Figura 17 – Cluster 16. Parque Natural Municipal de Petrópolis.....	32

## LISTA DE TABELAS

Tabela 1 – Comparação entre as soluções .....	20
---	----

## SUMÁRIO

1 INTRODUÇÃO.....	11
2 FUNDAMENTAÇÃO TEÓRICA .....	13
2.1 Clusterização.....	13
2.1.1 Medidas de similaridade.....	14
2.1.2 Algoritmo DBSCAN .....	15
2.2 POIs (Pontos de interesse) .....	16
2.3 Distribuição espacial de fotos .....	16
2.4 <i>Bounding Box</i> (Caixa de contorno).....	17
3 TRABALHOS RELACIONADOS .....	18
4 PROCEDIMENTOS METODOLÓGICOS .....	21
4.1 Coleta e preparação dos dados.....	22
4.2 Escolha e configuração do software de clusterização.....	23
4.3 Clusterização e Análise dos dados.....	24
4.4 Elaboração da amostragem e validação dos dados .....	24
5 RESULTADOS .....	25
5.1 Experimento.....	25
5.2 Mapas.....	25
5.3 Validação dos resultados .....	27
5.3.1 Cidade do Rio .....	27
5.3.2 Cidade de Petrópolis.....	30
6 CONSIDERAÇÕES FINAIS .....	33
REFERÊNCIAS .....	34
APÊNDICE A - SCRIPT UTILIZADO PARA <i>DOWNLOAD</i> DOS METADADOS DO SITE DE COMPARTILHAMENTO <i>FLICKR</i> .....	36
APÊNDICE B - SCRIPT UTILIZADO PARA JUNTAR OS ARQUIVOS XML EM APENAS UM.....	37

## 1 INTRODUÇÃO

Quando as pessoas vão viajar e o assunto é a escolha do destino turístico, alguns demoram a encontrar o local que consideram mais apropriado para seus momentos de lazer. Escolher um dos locais mais bonitos não é uma tarefa fácil quando há inúmeras opções disponíveis. Em muitos casos, os turistas fazem uso da recomendação de amigos que já frequentaram o lugar no qual tem interesse, através das fotos colocadas em sites de compartilhamento.

Com o uso das redes sociais, os usuários estão cada vez mais publicando as imagens de seus locais visitados, fazendo com que essas fotos tenham um valor notório para a pesquisa de identificação de pontos turísticos. Segundo Yang, Gong e U (2011), na Web 2.0, as pessoas partilham facilmente diversos tipos de recursos com outras pessoas. As fotos são tiradas, postadas e qualquer pessoa pode ou poderá ver e baixar a informação para o seu computador. Com isso, tornou-se mais fácil saber onde as pessoas estão e conseguir recomendações de locais interessantes. “O pressuposto é que muitas fotos tiradas perto umas das outras tendem a indicar pontos de interesse” (LACERDA et al., 2013).

Quaisquer fotos publicadas na Internet possuem dados que são armazenados nos repositórios de fotos disponíveis na *web*. Essas informações chamadas de metadados são colocadas nos vários sites de compartilhamento de foto ao realizar o *upload*, uma vez que essas são armazenadas diretamente no arquivo de imagem, seguindo padrões como *exif* - quando os arquivos são criados pela câmera, ou *XMP* - os metadados são armazenados na imagem ou em um arquivo à parte (PHOTOMETADATA, 2015). “Em muitos serviços de álbuns web, as pessoas não só compartilham as fotos, mas também fornecem informações relevantes, tais como *tags*, títulos, descrição, instante de tempo e localização” (YANG, GONG e U, 2011).

Estas informações podem ser utilizadas tanto para análise de pontos turísticos como para outros fins, a fim de realizar pesquisas, como a descoberta de *POIs* (Pontos de interesse) de turistas em alguns locais, entre eles, pontos turísticos, comércio, pontos religiosos, hotéis, pousadas, etc. Os pontos de interesse (*POIs*) são locais de compartilhamento de gostos em comum de usuários. O problema em descobrir *POIs* através de fotos é definir o método de descoberta dos pontos de interesse. Segundo Yang, Gong e U (2011), o ponto de interesse é caracterizado por fotos com base nos metadados, como o tempo e localização geográfica. A partir dos dados armazenados nas fotos, como latitude, longitude, data, etc, é possível saber a localização exata de onde a foto foi tirada, bem como o dia e a

hora, podendo assim determinar a partir de outras fotos tiradas por vários usuários, se aquele é um ponto de interesse em comum entre os usuários daquele site de compartilhamento; caracterizando um ponto turístico.

Uma das abordagens utilizadas na problemática de descoberta de pontos de interesse por densidade espacial é a clusterização. Clusterização é a classificação não-supervisionada de dados, formando agrupamentos ou clusters (JAIN, MURTHY e FLYNN, 1999). O princípio básico é que, dados com padrões de similaridade em um cluster, são similares entre si, do que com dados de outros clusters (JAIN, MURTHY e FLYNN, 1999). Algumas ferramentas usadas para clusterização de dados como *RapidMiner* (RAPIDMINER, 2015), *R*, *Weka* (HALL et al., 2009) e *Elki* (ELKI, 2015), permitem a realização de agrupamento sobre dados similares.

Este trabalho propõe, a partir de informações de *geotags* – informações de localização geográfica anexadas a fotos - armazenadas nas fotos do site de compartilhamento *Flickr*<sup>1</sup>, identificar destinos turísticos de cidades do estado do Rio de Janeiro a partir de locais com maior densidade de fotos, caracterizando assim um destino turístico popular e um ponto de interesse entre os usuários da rede social. Alguns trabalhos na literatura atualmente, tratam da clusterização de fotos para descoberta de *POIs*, como Lacerda et al. (2014) e Brilhante et al. (2013). Este trabalho realiza a identificação de destinos baseada nas escolhas dos usuários da rede social *Flickr*, ajudando turistas na escolha de destinos para possíveis viagens.

O trabalho está estruturado da seguinte maneira: A próxima seção aborda os conceitos relacionados ao trabalho, como Clusterização, Distribuição Espacial e pontos de interesse (*POIs*). Em seguida, apresenta-se os trabalhos utilizados como base da pesquisa. Depois, explica-se a execução detalhada do trabalho e as ferramentas utilizadas no experimento. Posteriormente, relata-se os resultados obtidos com o experimento. Por último, as considerações finais e definição de trabalhos futuros.

---

<sup>1</sup> <https://www.flickr.com>

## 2 FUNDAMENTAÇÃO TEÓRICA

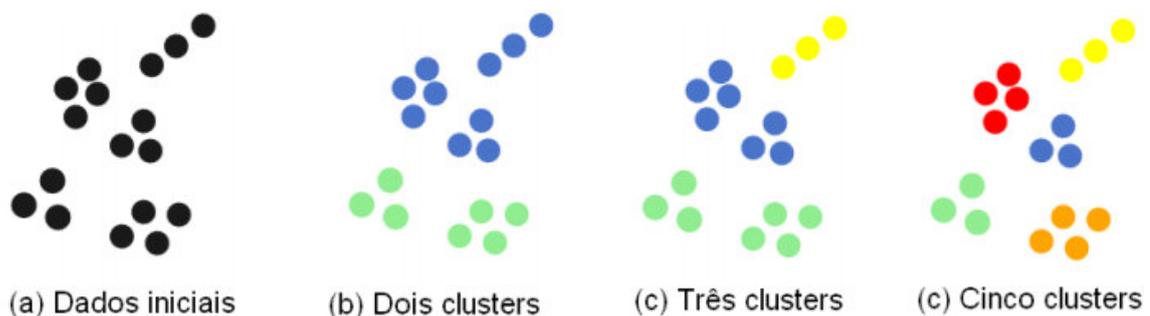
Nessa seção será apresentada uma visão geral sobre os conceitos-chave deste trabalho. Nas seções 2.1, 2.2 e 2.3 são apresentados os conceitos relacionados à clusterização, softwares de clusterização, POIs (pontos de interesse), distribuição espacial das fotos e bounding box, que foram a base para o projeto.

### 2.1 Clusterização

Clusterização é a descoberta de pontos de similaridade em um conjunto de dados, que são chamados “clusters”. Segundo Jain, Murthy e Flynn (1999), a análise de clusterização é a organização de uma coleção de padrões em grupos com base na similaridade. É muito utilizada em processos onde há pouco conhecimento sobre a base de dados.

Há dois tipos de análise de similaridade de dados: classificação supervisionada (análise discriminante) e classificação não supervisionada (clusterização). Na análise supervisionada, existem padrões de classificação pré-formados para o agrupamento dos dados. Na classificação não-supervisionada não há padrões de classificação e o agrupamento pode ser feito de inúmeras formas, de acordo com os dados existentes (JAIN, MURTY e FLYNN (1999)). Na Figura 1, temos o exemplo de clusterização com 2, 3 ou 5 clusters.

**Figura 1:** Exemplo de clusterização



Fonte: Oliveira, (2008).

Neste trabalho, os dados da clusterização foram identificados por densidade. A técnica de agrupamento irá demonstrar onde as coordenadas retiradas das fotos são similares e acontecem com maior intensidade, formando assim um cluster. Existem inúmeros algoritmos que tratam de dados a serem clusterizados. Alguns, como o DBSCAN tratam de dados de

clusters por densidade. K-means é um algoritmo que trabalha para minimizar os pontos centrais de distância entre todos os clusters (KRIEGEL et al., 2012).

A diferença entre os dois algoritmos é que DBSCAN trabalha com a descoberta de clusters com parâmetros sensíveis determinados pelo usuário, além de trabalhar com a descoberta de clusters através da densidade de pontos. DBSCAN, em comparação com K-means trabalha com a descoberta de clusters, deixando o outro algoritmo em desvantagem por não precisar de número de clusters previamente definidos. K-means trabalha com a quantidade de clusters já determinada, executando a descoberta da melhor forma de dividir os  $k$  grupos definidos. “Cada cluster é representado por um dado, chamado centróide do cluster. Sua função busca minimizar a variação entre os dados em relação ao centróide de cada cluster e é eficiente principalmente quando os clusters estão isolados” (OLIVEIRA, 2008). Cada centróide de cada cluster é calculado várias vezes até que se tenha uma melhor distribuição dos dados entre todos os clusters.

Mesmo com os algoritmos de clusterização já existentes, ainda há alguns valores que não se encaixam em nenhum cluster, por não possuir similaridade com os outros dados da coleção. Esses dados, chamados *outliers*, são dados incompletos ou que não possuem nenhuma característica em comum com outros. Pontos *outliers* são pontos atípicos ou inconsistentes que, por não pertencerem a cluster nenhum, são descartados (ESTER et al., 1996).

### 2.1.1 Medidas de similaridade

Dados para a formação de clusters baseados em densidade são geralmente organizados pelo nível de proximidade entre os pontos em um certo espaço. “Quando itens (unidades ou casos) são clusterizados, sua proximidade é indicada por algum tipo de distância. Por outro lado, as variáveis são agrupadas baseadas no seu coeficiente de correlação ou outras medidas estatísticas de associação” (KASNAR e GONÇALVES, 2009).

A escolha de uma medida de similaridade é crucial para a análise do cluster, especialmente para um tipo particular de algoritmos de agrupamento. Existem várias medidas de similaridade para análise. Alguns exemplos são Euclidiana, Coeficiente de Correlação de Pearson, e Similaridade Jaccard (MEYER, 2002).

Segundo Huang (2008), a medida de similaridade reflete o grau de proximidade e separação dos objetos alvo e deve corresponder às características que são acreditadas para distinguir os grupos incorporados nos dados. Em muitos casos, estas características são

dependentes dos dados ou do contexto do problema, e não há nenhuma medida que seja universalmente a melhor para todos os tipos de problemas de agrupamento.

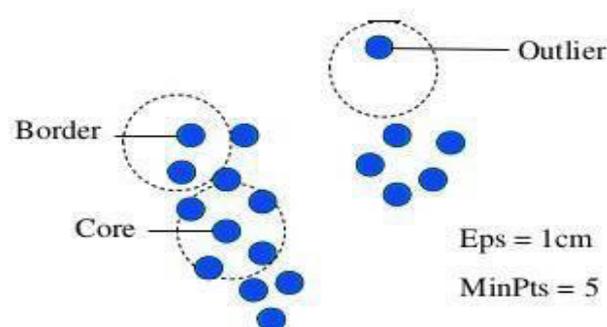
### 2.1.2 Algoritmo DBSCAN

DBSCAN (Clusterização baseada em densidade espacial de aplicações com ruído) é um algoritmo de clusterização que assume que clusters são regiões densas baseados no agrupamento de parâmetros similares de entrada. “ A ideia é que cada dado de um cluster tenha uma vizinhança de um determinado raio que contenha um número mínimo de pontos” (OLIVEIRA, 2008). O DBSCAN é um dos algoritmos mais utilizados na literatura. DBSCAN usa os mesmos parâmetros para definir todos os clusters. Os parâmetros de densidade do cluster "menos denso" são bons candidatos para os valores que especificam a densidade mais baixa e o que não é considerado como sendo ruído (ESTER et al., 1996).

Para se iniciar qualquer processo com DBSCAN, é necessário definir dois parâmetros:

- **Eps:** Define o raio máximo da vizinhança, ou seja, a distância máxima entre dois pontos para serem considerados vizinhos. (ESTER et al., 1996)
- **MinPts:** Densidade mínima, ou seja, número mínimo de dados para serem formados clusters a partir da expansão de um objeto.

**Figura 2:** Funcionamento do DBSCAN



Fonte: Elaborada pelo autor (adaptada de Han et al. 2013)

Utilizando a Figura 2 como exemplo temos: dado um ponto P, baseado nos parâmetros de *Eps* e *MinPts*, define-se quais pontos estão dentro do raio definido pelo parâmetro. Se P for um ponto central (*Core point*), ou seja, se houver pontos suficientes mediante o parâmetro *MinPts* alcançáveis por seu raio, um *cluster* é formado. Se P for um

ponto de borda (*Border point*), então não haverá pontos alcançáveis por sua densidade. Sendo assim, visita-se o próximo ponto e o processo é repetido até que não haja mais pontos dentro do raio. Se o raio de P não alcançar nenhum ponto e for totalmente fora de vista dos outros dados, P é um *outlier* (ESTER et al., 1996).

## 2.2 POIs (Pontos de interesse)

Os pontos de interesse são locais caracterizados por dados de coordenadas geográficas, utilizados para identificação de destinos, sejam eles turísticos ou pontos de coordenadas de GPS. São pontos ou locais de interesse em comum entre pessoas.

Mais utilizado na cartografia, são caracterizados por dados que contém informações de coordenadas geográficas de um determinado lugar. Com relação a atividades de fotos em sites de compartilhamento, Yang, Gong e U (2011) citam que, o ponto de interesse pode ser caracterizado por atividades com base em metadados como o tempo e localização geográfica.

## 2.3 Distribuição espacial de fotos

Fotos podem ser capturadas em diversos momentos e lugares. Algumas câmeras fotográficas registram no momento da captura, informações como coordenadas geográficas, tamanho da foto, modelo da câmera, resolução, etc. A partir dessas informações, ao fazer o *upload* da imagem, os dados são colocados em conjunto com a foto.

Com a utilização de imagens com geolocalização, é possível identificar os pontos dos locais de interesse exatos, utilizando fotos capturadas em comum com inúmeros turistas.

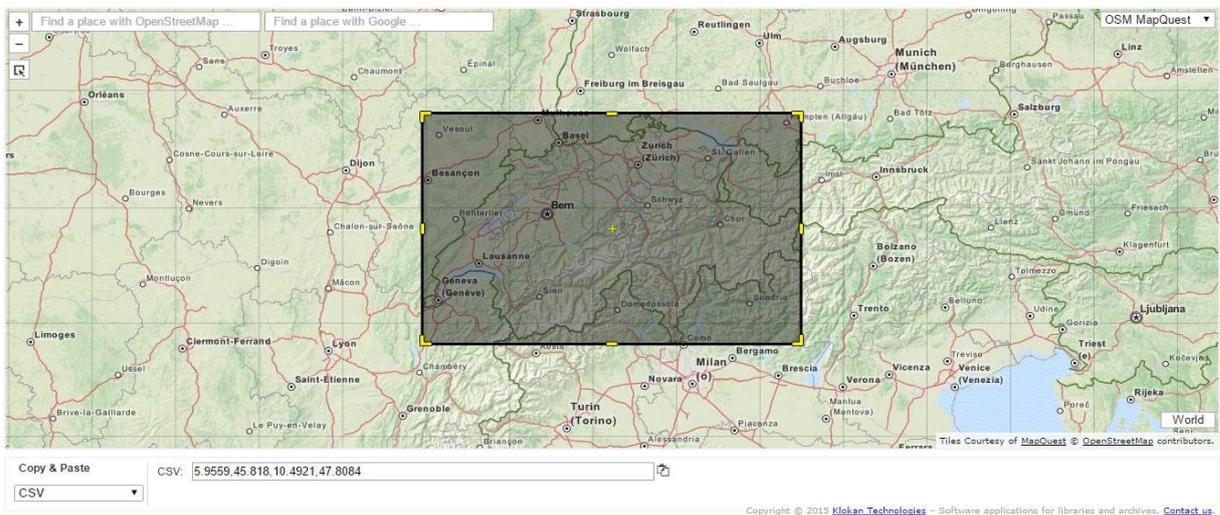
As informações de geolocalização podem fornecer a base para muitas outras tarefas de captar informações na imagem, tais como estimar a densidade populacional, estimativa de cobertura do solo ou classificação urbano/rural (HAYS e EFROS, 2008).

Ainda há outra funcionalidade para as informações geográficas das fotos. A partir dos metadados capturados, é possível filtrar as informações para definir, além também, o aglomerado de pessoas no local naquele momento, que tipo de pessoas frequentam aquele ponto, entre outras informações.

## 2.4 Bounding Box (Caixa de contorno)

*Bounding Box* é, em softwares de ilustração e paginação, uma caixa retangular que pode ser colocada sobre um objeto selecionado; é formada por quatro ou oito pontos de controle que são manipulados para distorcer, mover, girar etc. No contexto geográfico, é um campo de mínimos e máximos pares de coordenadas geográficas que formam uma área quadrada de seleção em um mapa. Na Figura 3, um exemplo de gerador de *bounding box* disponível na internet.

**Figura 3:** Gerador de *Bounding Boxes*



Fonte: Bounding Box<sup>2</sup>

<sup>2</sup> <http://boundingbox.klokantech.com/>

### 3 TRABALHOS RELACIONADOS

Clusterização de fotos geomarcadas é um tema explorado em trabalhos acadêmicos de alguns autores como (CRANDALL et al., 2009), (LACERDA et al., 2014), (BRILHANTE et al., 2013) e (YANG, GONG e U, 2011).

Crandall et al. (2009) fazem uma análise a partir de fotos com *geotags* coletadas do Flickr. A partir do local em que as fotos de várias pessoas foram obtidas pôde-se estabelecer uma estrutura relacional que serviu para definir quais tipos de fotos foram tiradas em lugares populares. Após isso, foi pesquisada a interação, a estrutura de relação, o conteúdo das fotos e métodos que poderiam ser utilizados para prever os locais de características visuais; textuais e temporais das fotos. Segundo Crandall et al. (2009), os recursos visuais e temporais melhoram a capacidade de estimar a localização de uma foto, em comparação com o uso de recursos apenas textuais. Foram utilizadas essas técnicas para trabalhar com cerca de 35 milhões de imagens. Assim como Crandall et al. (2009), fizemos uma pesquisa para definir pontos de fotos geomarcadas que são mais populares entre os turistas, utilizando como validador de dados, o *Google Maps*, embora a metodologia de utilizar recursos visuais e temporais não esteja no escopo do projeto; seria uma boa forma de embasamento, já que apenas os recursos textuais algumas vezes não são suficientes.

Lacerda et al. (2014), propõem um algoritmo para análise de agrupamento de fotos com arquitetura orientada a serviço chamado *PhoCA*. Esse algoritmo realiza todas as funções relativas à clusterização de fotos, como *download*, processamento, reconhecimento de padrões de *clusters* e análise de dados utilizando cinco algoritmos já existentes na literatura, entre eles, DBSCAN. *PhoCA* funciona em cinco módulos e pode ser acessado diretamente de outros serviços *web*. O módulo *Experimental* permite que seja utilizado um algoritmo de clusterização para uma coleção de dados. O módulo *Cluster Analyzer* é responsável pela sumarização das estatísticas dos resultados obtidos na clusterização da coleção. O módulo chamado de *Summary Statistics* é responsável por apresentar alguns dados simples de estatísticas além de mostrar o tamanho dos clusters finais. Além desses, ainda possui os módulos de *Autorização e Autenticação* e *Configuração* (LACERDA et al., 2012). Em seu teste de execução, utiliza o DBSCAN com os valores de  $MinPts = 30$  e  $\epsilon$  (Epsilon) = 100 metros, em conjunto com seu algoritmo *Compass Clustering*, proposto em Lacerda et al. (2012), para detecção de pontos de interesse. Assim como Lacerda et al. (2013), este trabalho irá realizar uma forma de clusterização utilizando uma base de dados constituída de metadados com *download* manual baseada no algoritmo DBSCAN, utilizando os mesmos

parâmetros usados no trabalho de Lacerda et al. (2014), sem o algoritmo de agrupamento *Compass Clustering*.

Brilhante et al. (2013), propõem um algoritmo chamado *TripBuilder*. Retirando do *Flickr* as fotos de inúmeros turistas, combinaram os itinerários dos visitantes com alguns pontos de interesses disponíveis na *Wikipedia* e assim disponibilizaram sugestões de lugares para serem visitados por esses turistas. “*TripBuilder* toma como entrada o destino pretendido, o tempo disponível para a visita, o perfil do usuário (explícito ou implícito), e constrói um *tour* personalizado atravessando uma seleção de POIs” (BRILHANTE et al., 2013). A partir dessas informações, o algoritmo propõe um itinerário com visitas a pontos turísticos no tempo disponível pelos usuários, utilizando também o mapa para definir o melhor caminho. “O passeio recomendado maximiza os interesses do usuário e respeita a restrição de tempo de visita, uma vez que leva em conta tanto o tempo para desfrutar das atrações quanto o tempo necessário para a movimentação de um *POI* para o próximo” (BRILHANTE et al., 2013). Os resultados experimentais da execução do algoritmo em três cidades turísticas diferentes da Europa mostraram que o uso do *TripBuilder* é eficaz e superou outros algoritmos. Assim como Brilhante et al. (2013), trabalharemos com identificação de destinos turísticos utilizando a clusterização para a descoberta de *POIs*, através de metadados de fotos do *Flickr*.

Yang, Gong e U (2011), utilizam o algoritmo de agrupamento espectral para identificar POIs. O trabalho se resume em conseguir realizar uma forma de clusterização auto-ajustável, em que os parâmetros são adaptados automaticamente de forma adequada para a execução. “Não há uma abordagem unificada para atribuir os parâmetros de clusterização; especialmente por as características de POIs serem imensamente variadas em diferentes metrópoles e locais” (YANG, GONG e U, 2011). Com metadados coletados do *Flickr*, o método foi testado com clusterização baseada na data e informações geoespaciais. Então, o objetivo do experimento era descobrir POIs de alta densidade utilizando o método proposto. Assim como neste trabalho, utilizamos a clusterização para a descoberta de *POIs*, embora não usando o algoritmo auto-ajustável, e sim DBSCAN, que necessita de alguns parâmetros para a execução.

**Tabela 1:** Comparação entre as soluções

<b>Trabalho</b>	<b>Algoritmo</b>	<b>Recursos</b>
Crandall et al. (2009)	Clusterização espectral	Fotos do <i>Flickr</i> (visuais, temporais e textuais)
Lacerda et al. (2014)	k-means, mean shift, DBSCAN, clusterização espectral	Fotos do <i>Flickr</i> ( <i>geotags</i> )
Brilhante et al. (2013)	DBSCAN	Fotos do <i>Flickr</i> ( <i>geotags</i> ) Paginas da Wikipedia
Yang, Gong e U (2011)	DBSCAN	Fotos do <i>Flickr</i> ( <i>geotags</i> )
Este trabalho	DBSCAN	Fotos do <i>Flickr</i> ( <i>geotags</i> )

Fonte: Elaborada Pelo autor

## 4 PROCEDIMENTOS METODOLÓGICOS

O projeto aborda uma solução para descobrir destinos turísticos baseada na densidade de fotos geomarcadas com similaridade de localização. Em um primeiro momento, foi feito o estudo da API do site *Flickr* para descobrir uma forma de realizar o *download* de fotos que possuíssem geomarcagem. Como o *Flickr* disponibiliza cerca de 500 metadados por página, através de um dos métodos da API chamado *bbox*, que cria uma caixa delimitadora com dois pares de coordenadas para limitar o *download*, foram baixados cerca de 430.000 metadados, cerca de 280 mb de arquivos.

**Figura 4:** Página da API Explorer do *Flickr*

The screenshot displays the Flickr API Explorer interface. At the top, there are navigation links for 'flickr', 'Cadastrar-se', 'Explorar', and 'Criar'. A search bar contains the text 'Fotos, pessoas ou grupos'. Below this, there are three rows of form fields: 'extras' (optional, checked, value 'geo'), 'per\_page' (optional, checked, value '500'), and 'page' (optional, unchecked, empty). Underneath, the 'Saída' dropdown is set to 'XML (REST)'. There are two radio buttons: 'Assinar chamada sem token de usuário?' (unchecked) and 'Não assinar chamada?' (checked). A 'Call Method...' button is present. Below the form, there is a blue link: 'Voltar para a documentação flickr.photos.search'. The bottom portion of the image shows a preview of the XML response, which is a list of photo metadata including photo id, owner, secret, server, farm, and title.

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
  <photos page="1" pages="9" perpage="500" total="4206">
    <photo id="24462376909" owner="39331916@N02" secret="c63d0a7164" server="1484" farm="2" title="Cabo Frio" />
    <photo id="24192183063" owner="44420378@N04" secret="c658f28e57" server="1481" farm="2" title="Para pront" />
    <photo id="24192111143" owner="44420378@N04" secret="2281e9f2b3" server="1492" farm="2" title="Para pront" />
    <photo id="24451080869" owner="44420378@N04" secret="37dee522d" server="1719" farm="2" title="Tenho essa" />
    <photo id="24684081582" owner="39821953@N05" secret="c7d10ca9d6" server="1669" farm="2" title="OSC04635" />
    <photo id="24484620590" owner="64161176@N03" secret="276ceccd67" server="1616" farm="2" title="Parque da" />
    <photo id="24482312340" owner="39331916@N02" secret="972f04f6d9" server="1704" farm="2" title="Wildstyle" />
    <photo id="24612009162" owner="27230084@N04" secret="f6787937dd" server="1629" farm="2" title="Sempre qu" />
    <photo id="24403110740" owner="64161176@N03" secret="3403324106" server="1517" farm="2" title="Parque da" />
    <photo id="24330925759" owner="64161176@N03" secret="7ee2c69cd9" server="1553" farm="2" title="Parque da" />
    <photo id="24071666473" owner="64161176@N03" secret="0c31397960" server="1699" farm="2" title="Parque da" />
    <photo id="24672293256" owner="64161176@N03" secret="5fafc1885c" server="1626" farm="2" title="Parque da" />
    <photo id="24672291036" owner="64161176@N03" secret="f9ffb300d2" server="1674" farm="2" title="Parque da" />
    <photo id="24330819319" owner="64161176@N03" secret="e0bfff5376" server="1542" farm="2" title="Parque da" />
  </photos>
</rsp>
</ok>
```

Fonte: *Flickr*

Como se pode ver na Figura 4, os dados de metadados são disponibilizados e em conjunto, o link para *download*. Os arquivos possuíam o máximo de 500 fotos por página, totalizando cerca de 500 mil fotos, entre as cidades do Rio de Janeiro e Petrópolis. Após o *download* das fotos e filtragem dos dados em latitude e longitude, foram testados os *frameworks* de clusterização Elki e RapidMiner para executar o algoritmo e definir qual a melhor solução para efetuar o agrupamento. As escolhas foram baseadas em pesquisas na literatura, chegando os dois a ter uma boa aceitação por serem *Open Source* e rápidos no processamento. Definido o software, foram executados os experimentos, obtidos os resultados e comparados os dados com os pontos no *Google Maps*, como medida de validação.

## 4.1 Coleta e preparação dos dados

Com o estudo da API, foi visto que o *Flickr* gera uma chave com validade de 24h para manipulação dos dados. Foi programado um *script* em shell script, resultando no código configurado para baixar metadados em XML do site. Como o *Flickr* disponibiliza cerca de 500 metadados por página, através de um dos métodos da API chamado *bbox*, que cria uma caixa delimitadora com dois pares de coordenadas para limitar o *download*; foram baixadas somente as fotos das cidades do Rio de Janeiro (400.000) e Petrópolis (8.000) para o processamento. Esse foi o total de fotos disponível no site na data do *download*. Além disso, no intuito de obter a localização espacial, foi adicionado o parâmetro *geo*, o qual possibilitou incluir no arquivo XML as coordenadas de latitude e longitude onde cada foto foi tirada. Na Figura 5, temos um exemplo de arquivo XML com os parâmetros *geo* delimitados por uma *bounding box*:

**Figura 5:** Arquivo XML retirado do *Flickr*

---

```

- <rsp stat="ok">
- <photos page="1" pages="546" perpage="500" total="272639">
  <photo id="16960107844" owner="93145299@N08" secret="83c2618aa5" server="8888" farm="9" title="Santa Ceia 12 de Maio 2015" ispublic="1"
  isfriend="0" isfamily="0" datetaken="2015-05-12 19:53:21" datetakengranularity="0" datetakenunknown="0" latitude="-9.484386" longitude="-
  35.852465" accuracy="16" context="0" place_id="8i_DqQRVV73pBa4" woeid="457945" geo_is_family="0" geo_is_friend="0" geo_is_contact="0"
  geo_is_public="1"/>
  <photo id="17395078090" owner="93145299@N08" secret="5241ace432" server="8896" farm="9" title="Santa Ceia 12 de Maio 2015" ispublic="1"
  isfriend="0" isfamily="0" datetaken="2015-05-12 19:54:21" datetakengranularity="0" datetakenunknown="0" latitude="-9.484386" longitude="-
  35.852465" accuracy="16" context="0" place_id="8i_DqQRVV73pBa4" woeid="457945" geo_is_family="0" geo_is_friend="0" geo_is_contact="0"
  geo_is_public="1"/>
  <photo id="17395068060" owner="93145299@N08" secret="b0eea11f1c" server="8823" farm="9" title="Santa Ceia 12 de Maio 2015" ispublic="1"
  isfriend="0" isfamily="0" datetaken="2015-05-12 19:53:47" datetakengranularity="0" datetakenunknown="0" latitude="-9.484386" longitude="-
  35.852465" accuracy="16" context="0" place_id="8i_DqQRVV73pBa4" woeid="457945" geo_is_family="0" geo_is_friend="0" geo_is_contact="0"
  geo_is_public="1"/>
  <photo id="17582541781" owner="93145299@N08" secret="26d58d48b0" server="8889" farm="9" title="Santa Ceia 12 de Maio 2015" ispublic="1"
  isfriend="0" isfamily="0" datetaken="2015-05-12 20:01:14" datetakengranularity="0" datetakenunknown="0" latitude="-9.484386" longitude="-
  35.852465" accuracy="16" context="0" place_id="8i_DqQRVV73pBa4" woeid="457945" geo_is_family="0" geo_is_friend="0" geo_is_contact="0"
  geo_is_public="1"/>
  <photo id="17580250512" owner="93145299@N08" secret="ca4c0db282" server="5329" farm="6" title="Santa Ceia 12 de Maio 2015" ispublic="1"
  isfriend="0" isfamily="0" datetaken="2015-05-12 20:04:22" datetakengranularity="0" datetakenunknown="0" latitude="-9.484386" longitude="-
  35.852465" accuracy="16" context="0" place_id="8i_DqQRVV73pBa4" woeid="457945" geo_is_family="0" geo_is_friend="0" geo_is_contact="0"
  geo_is_public="1"/>

```

Fonte: Elaborada pelo autor

A partir do momento em que os *metadados* estavam disponíveis, foi realizada uma transformação dos dados que seriam usados no experimento. Foram retirados os elementos de latitude e longitude, colocados como entrada em um software de conversão, transformando-os em pontos (X,Y) do plano cartesiano e colocando-os em um arquivo csv para ser processado pelo software de clusterização.

## 4.2 Escolha e configuração do software de clusterização

Os softwares candidatos para o experimento foram *RapidMiner* e *Elki*.

*Elki* é software de mineração de dados de código aberto (AGPLv3) escrito em Java. Abrange, por um lado, uma coleção profunda e facilmente extensível de algoritmos para aplicações de mineração de dados, tais como clusterização, classificação e detecção de *outlier*, e por outro lado *Elki* incorpora e suporta estruturas de índice arbitrárias para suportar até mesmo grandes conjuntos de dados de alta dimensão (ACHTERT, KRIEGEL e ZIMEK, 2008).

*RapidMiner Studio* é um software de código livre para a concepção de processos analíticos avançados com aprendizagem de máquina, mineração de dados, mineração de texto, análise preditiva e análise de negócios (RAPIDMINER, 2015).

Os dados foram baixados no formato XML. Depois, através de um conversor de latitude e longitude, os dados foram transformados em pontos X,Y do plano cartesiano. Para o experimento com o algoritmo DBSCAN, foram utilizados parâmetros de pontos X e Y do plano cartesiano para o software de clusterização. O DBSCAN possui uma alta utilização na literatura, por ter uma boa eficiência se tratando de clusterização baseada em densidade. Foram utilizados os parâmetros baseados no trabalho de Lacerda et al. (2012), que definiu os parâmetros de  $Minpts = 30$  e  $\epsilon = 100$  metros.

A máquina utilizada possui as seguintes características:

- 8GB de memória RAM (No qual foi utilizado 2GB para a clusterização dos dados)
- Processador Core i5

Em um primeiro momento, foi feito um teste com o *RapidMiner*. A versão *Free* do programa permitia que fosse utilizado 1GB de RAM para a execução. O teste mostrou que *RapidMiner* é inviável, demorando mais de 12h para processar o arquivo total. Por isso, *RapidMiner* foi descartado. O teste com *Elki* foi feito também com uma amostra do arquivo total do experimento. O software se mostrou rápido e eficaz no processamento, executando o procedimento com 2GB de RAM em apenas 30 minutos.

### 4.3 Clusterização e Análise dos dados

As execuções no Elki foram feitas no modo texto, com os parâmetros utilizados abaixo:

- -dbc.in: Usado para importar o arquivo de extensão .csv para o software.
- -algorithm: Seleção do algoritmo usado na execução, nos testes feitos, o DBSCAN.
- -resulthandler: Definido como *ResultWriter* para que a saída fosse no formato texto.

### 4.4 Elaboração da amostragem e validação dos dados

Após a clusterização, foram gerados mapas no software de informações geográficas QGIS<sup>3</sup> para ilustrar a expansão dos clusters e sua localização. Cada cluster foi pintado de uma cor para ilustração. Os *outliers* foram pintados de cor diferente para se destacar dos demais clusters.

Para a validação dos dados, foram feitas pesquisas manuais pelos pontos de algumas amostras dos cluster no *Google Maps*<sup>4</sup> e, após isso, tiradas cópias das imagens obtidas dos pontos turísticos. Alguns pontos se encontram a poucos metros do ponto marcado pelo *Google Maps* representando o ponto turístico.

---

<sup>3</sup> <http://www.qgis.org/en/site/>

<sup>4</sup> <https://www.google.com/maps>

## 5 RESULTADOS

Com a base de dados dos metadados do *Flickr*, foram detectadas as presenças de pontos turísticos, além de pontos como escolas, faculdades e ruas. Com os mapas gerados no QGIS, os pontos foram descobertos e dispostos ao redor do mapa. A partir dos clusters gerados, foi feita uma validação manual com o *Google Maps* para identificar os *POIs*.

### 5.1 Experimento

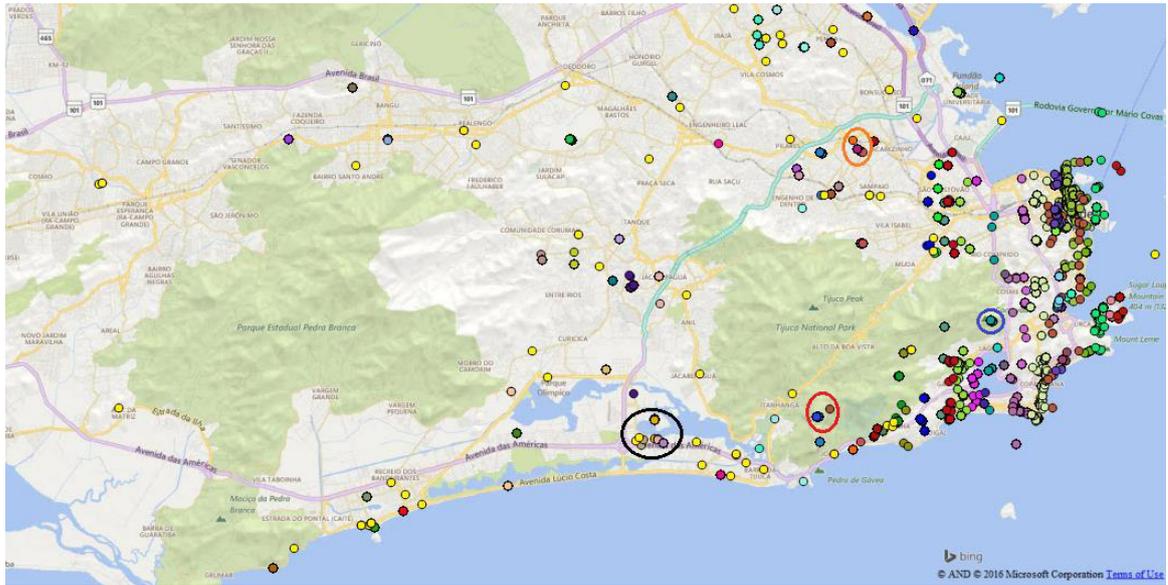
Na pesquisa de escolha das cidades, foram comparadas bases de dados do *Flickr* de inúmeras cidades do país. Devido aos dados para outros lugares serem escassos, foram escolhidas duas cidades turísticas do estado do Rio de Janeiro com tamanhos diferentes e uma base de dados razoável.

No experimento baseado nos parâmetros usados por Lacerda et al. (2012), o  $\epsilon$  foi definido em 100 metros e o *MinPts* definido em 30, foram gerados 66 *clusters* na execução dos dados com localização no Rio de Janeiro em aproximadamente 30 minutos de execução e 17 clusters gerados na execução referente a cidade de Petrópolis com 5 minutos de execução. A execução foi realizada no *Elki* em modo texto.

### 5.2 Mapas

Os mapas gerados pelos clusters das cidades foram gerados no software de informações geográficas QGIS. Cada cor representa um cluster diferente. O mapa da Figura 6 representa o resultado da clusterização dos pontos no Rio de Janeiro. Alguns clusters marcados estão identificados na tela com círculos. O círculo vermelho representa clusters situados no Parque Florestal da Tijuca. Os clusters em preto representam pontos na Lagoa da Tijuca. O cluster em azul representa o Corcovado e o círculo em laranja representa o Maracanã e suas proximidades. Os pontos marcados em amarelo representam os *outliers*.

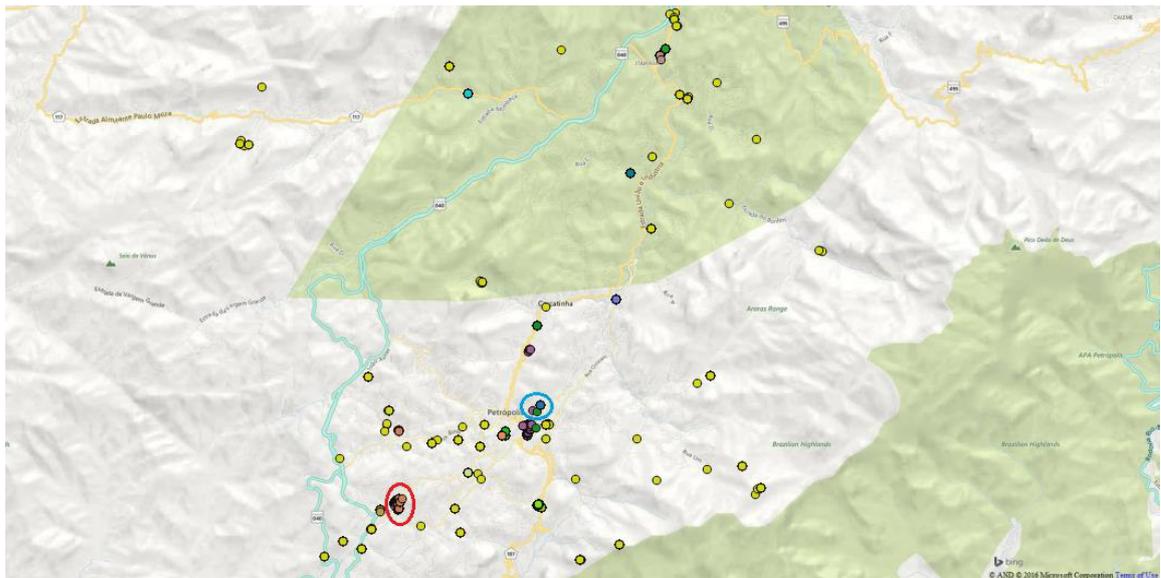
**Figura 6:** Representação dos clusters no Rio de Janeiro



Fonte: Elaborada pelo autor

O mapa da Figura 7 representa o resultado da clusterização no dos pontos na cidade de Petrópolis. O círculo em vermelho representa os clusters referentes ao Palácio Quitandinha. O círculo em azul representa, respectivamente, a Catedral São Pedro de Alcântara (roxo), o Museu Imperial (verde) e o Parque Natural Municipal de Petrópolis (azul). Os pontos marcados em amarelo representam os *outliers*.

**Figura 7:** Representação dos clusters na cidade de Petrópolis



Fonte: Elaborada pelo autor

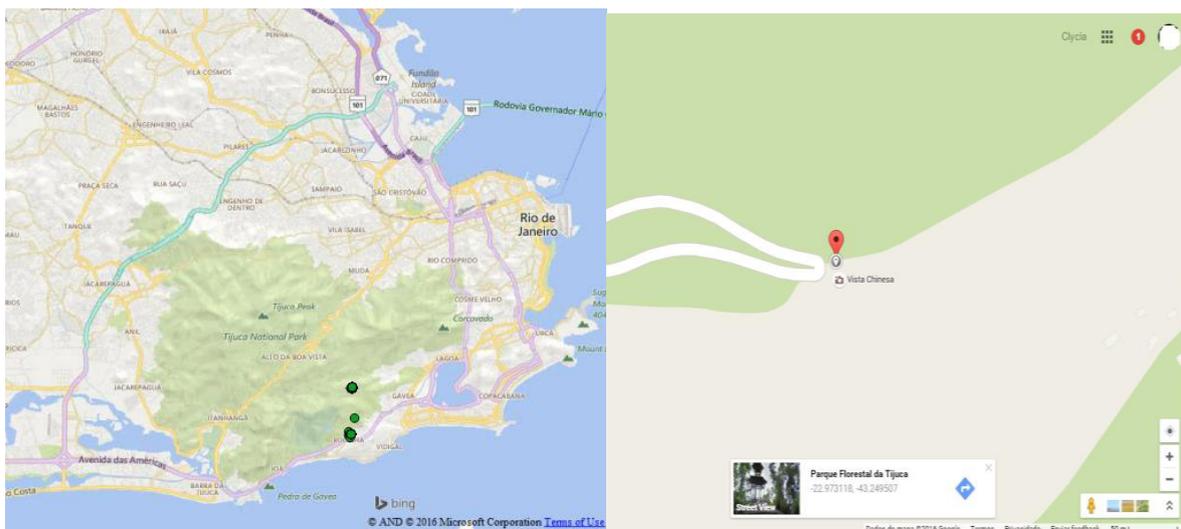
### 5.3 Validação dos resultados

A partir dos clusters gerados, alguns clusters foram selecionados para a validação dos dados e descoberta dos *POIs*. Foram colocados vários pontos do cluster para um melhor resultado. As coordenadas disponíveis foram testadas no *Google Maps*, e os *POIs* descobertos foram comparados com o local do ponto marcado. O *Google Maps* é um serviço de visualização de mapas e ruas desenvolvido pela empresa *Google*. Foram escolhidos 5 dos clusters da cidade do Rio de Janeiro e Petrópolis para validação.

#### 5.3.1 Cidade do Rio

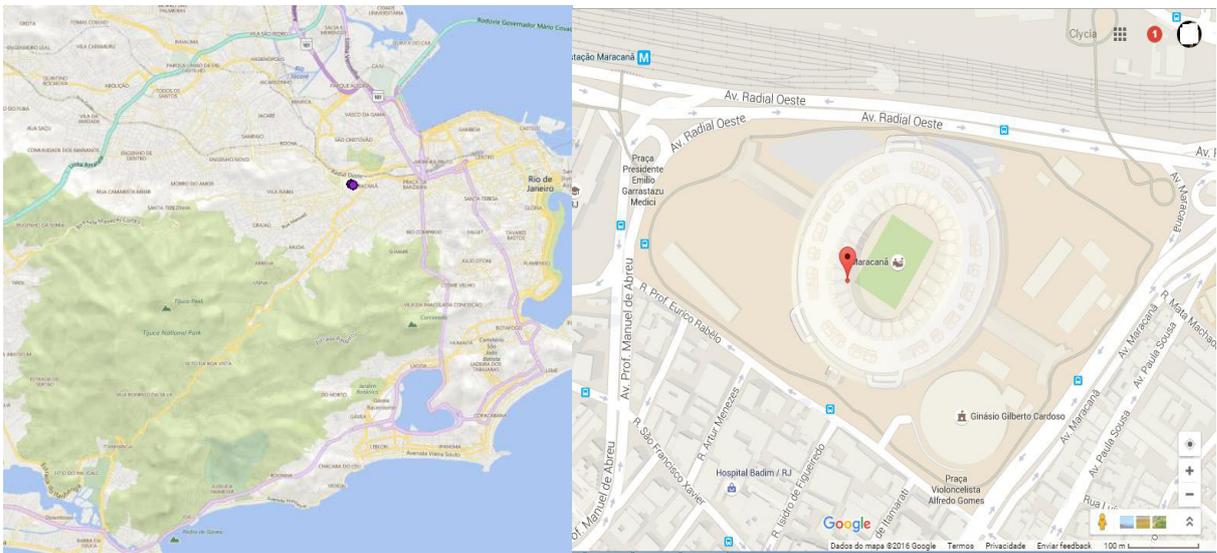
A cidade do Rio de Janeiro gerou 66 clusters a partir dos metadados coletados com 30 minutos de execução. Para validação dos dados, foram escolhidos 5 clusters aleatórios, cujas coordenadas foram colocadas no *Google Maps* para identificação do possível ponto de interesse. Nas Figuras 8, 9, 10, 11 e 12, podemos ver respectivamente o Parque Nacional da Tijuca, o estádio do Maracanã, a praia da Barra da Tijuca, o Jardim Botânico e o Cristo Redentor, que são pontos turísticos muito famosos na cidade do Rio e frequentada por muitos turistas em várias épocas do ano.

**Figura 8:** Cluster 6. Parque Florestal da Tijuca



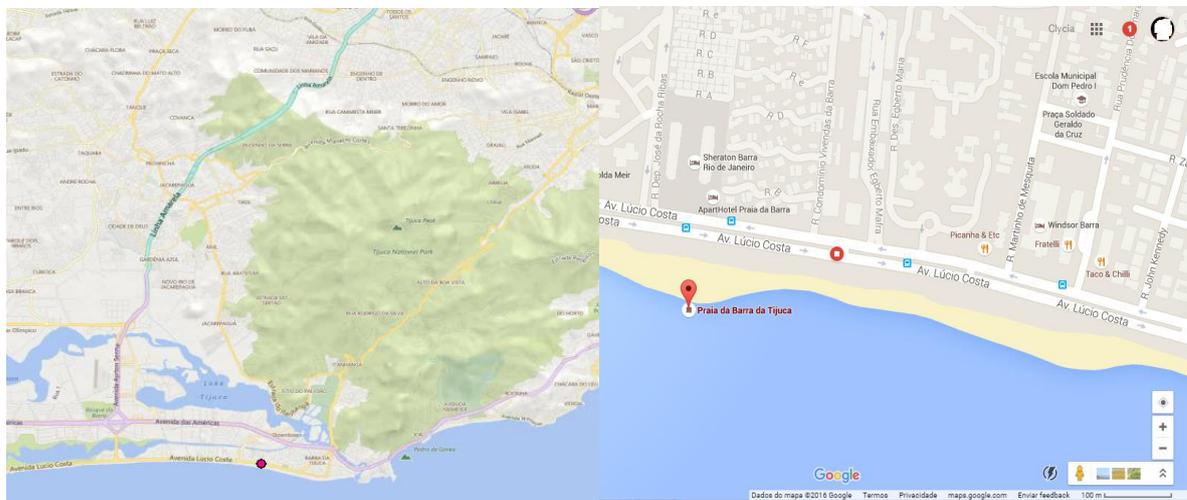
Fonte: Google Maps

**Figura 9: Cluster 5. Estádio Maracanã**



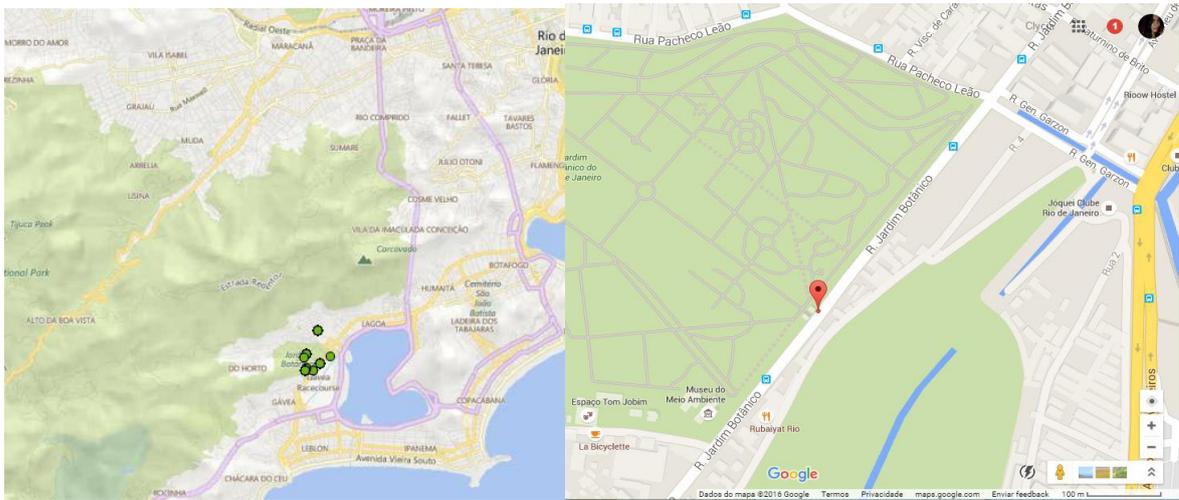
Fonte: Google Maps

**Figura 10: Cluster 28. Praia da Barra**



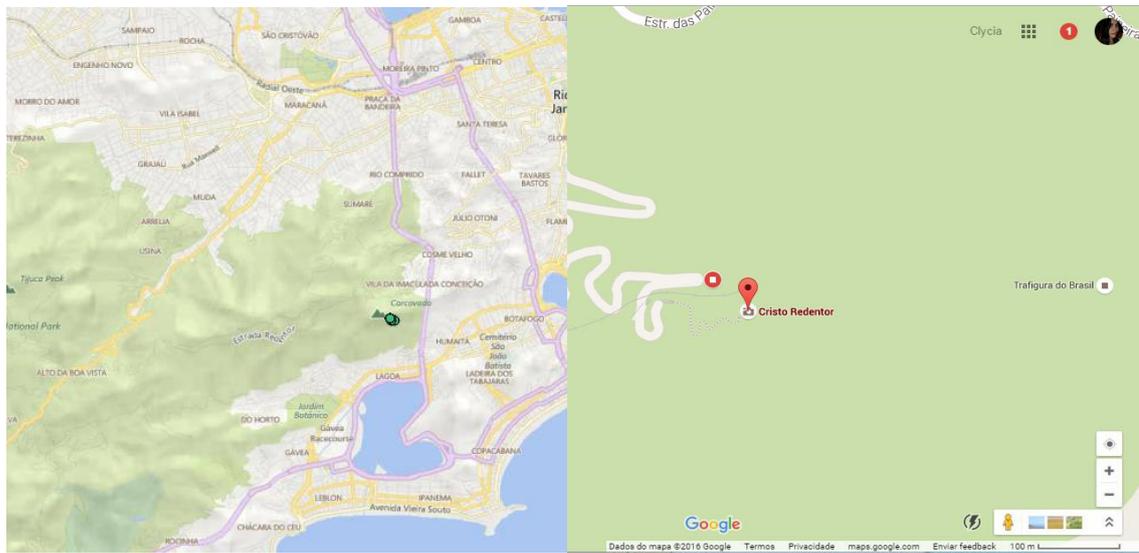
Fonte: Google Maps

**Figura 11: Cluster 37. Jardim Botânico**



Fonte: Google Maps

**Figura 12: Cluster 42. Cristo Redentor**

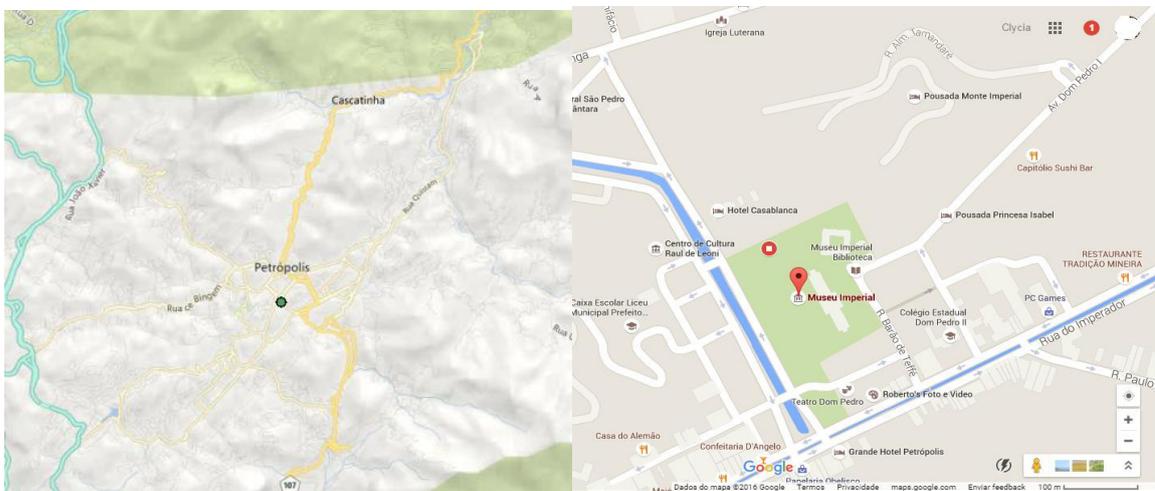


Fonte: Google Maps

### 5.3.2 Cidade de Petrópolis

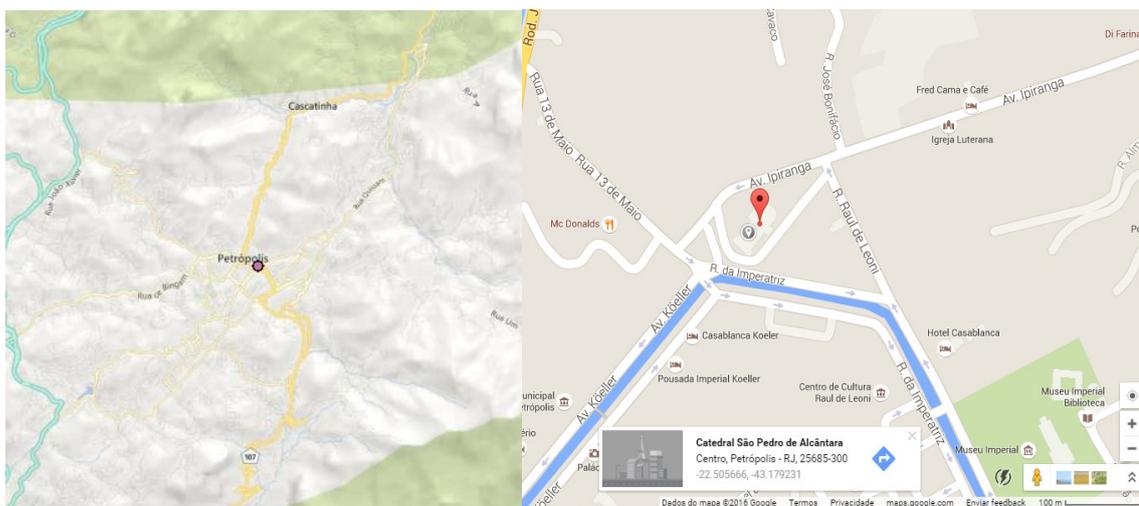
A cidade de Petrópolis gerou 17 clusters a partir dos metadados coletados. Para validação dos dados, foram escolhidos 5 clusters, cujas coordenadas foram colocadas no *Google Maps* para identificação do ponto de interesse. As Figuras 13, 14, 15, 16 e 17 representam respectivamente os pontos turísticos do Museu Imperial, Catedral São Pedro de Alcântara, Palácio Quitandinha, Praça da Liberdade e Parque Natural Municipal de Petrópolis. Todos são pontos turísticos muito visitados na cidade pelos turistas.

**Figura 13: Cluster1. Museu Imperial**



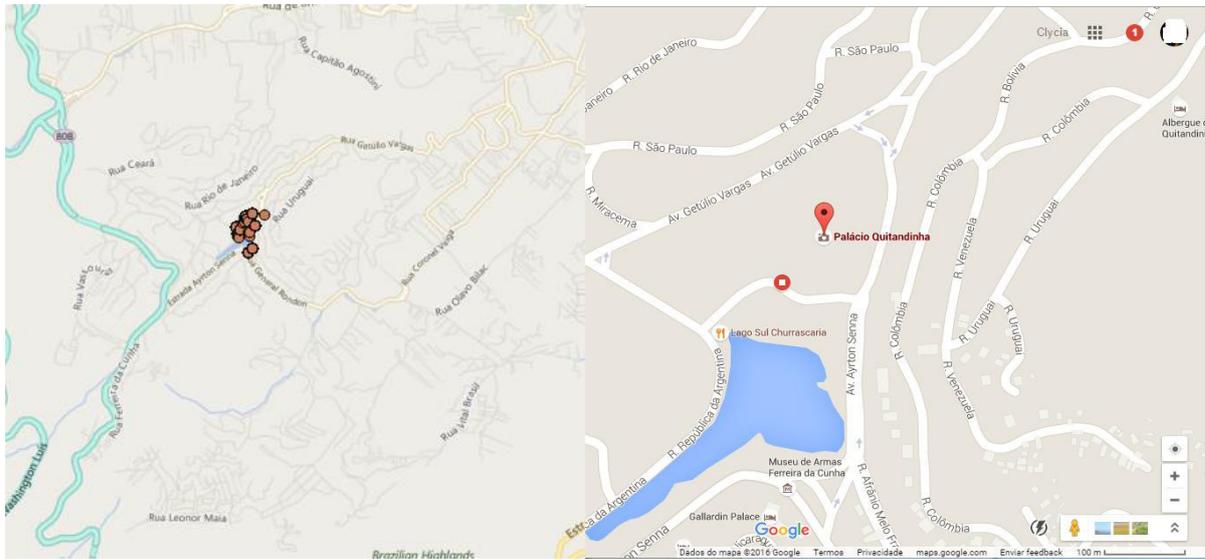
Fonte: Google Maps

**Figura 14: Cluster 2. Catedral São Pedro de Alcântara**



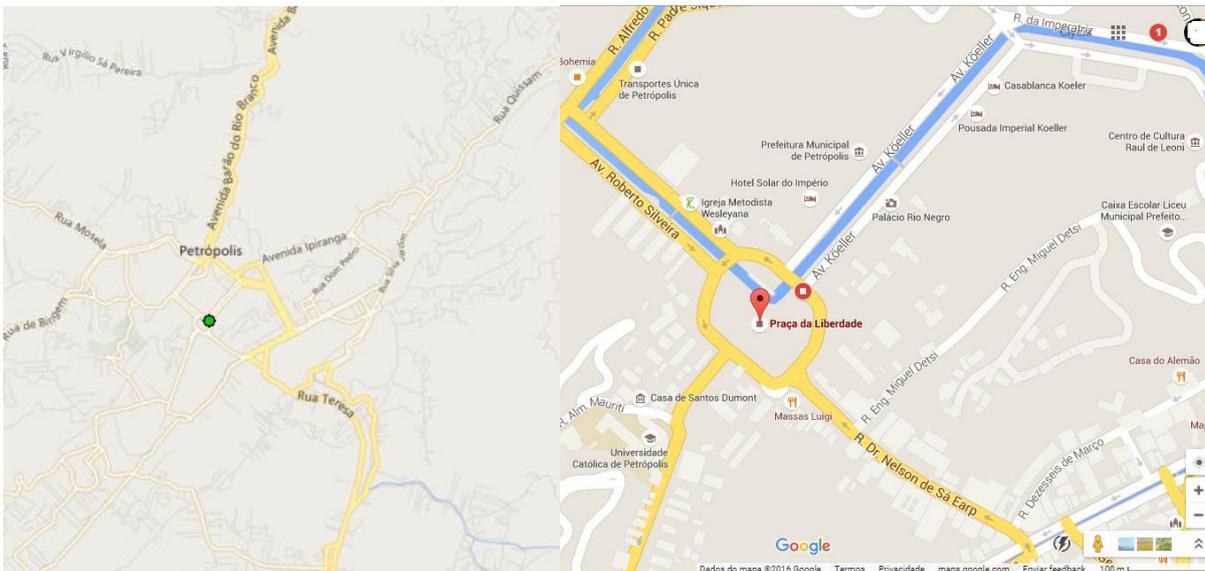
Fonte: Google Maps

**Figura 15: Cluster 8. Palácio Quitandinha**



Fonte: Google Maps

**Figura 16: Cluster 11. Praça da Liberdade**



Fonte: Google Maps



## 6 CONSIDERAÇÕES FINAIS

Este trabalho teve o objetivo de mostrar uma solução baseada em clusterização de fotos para a descoberta de destinos turísticos nas cidades do Rio de Janeiro e Petrópolis. A base de dados foi baixada, tratada através de scripts e nela aplicada ao algoritmo DBSCAN com os parâmetros utilizados por Lacerda et al. (2014), o qual foi possível detectar a presença de pontos turísticos, além de pontos como escolas, faculdades e ruas. A partir do mapa gerado no software de processamento de dados geográficos QGIS, distribuimos os pontos para descobrir sua localização e validamos os pontos manualmente a partir do site de visualização de mapas *Google Maps*.

Vale ressaltar que esse tipo de análise está cada vez mais popular, pela relevância que os metadados das fotos têm no sentido de descobrir as localizações de *POIs* de usuários para a área de identificação de destinos e trajetórias turísticas baseadas no agrupamento dos dados.

Como trabalhos futuros, podemos utilizar os valores de data e hora dos metadados para descobrir grupos de usuários que frequentam os mesmos pontos de interesses, bem como os mesmos horários, aplicando um algoritmo que trabalhe com outros parâmetros, e além disso, descobrir os horários em que os locais seriam mais frequentados pelas pessoas que utilizam a rede social para realizar a identificação. Também podemos validar e anotar os *POIs* a partir da proximidade dos clusters com pontos de interesse obtidos na *Wikipedia* ou outras fontes de dados geográficos. Além disso, pode-se trabalhar no desenvolvimento de uma aplicação *web* e/ou *mobile* para facilitar a identificação de pontos de interesse baseada na localização do usuário ou locais marcados pelo usuário em um mapa.

## REFERÊNCIAS

- BRILHANTE, Igo et al. **Where shall we go today?: Planning Touristic tours with TripBuilder**. In: PROCEEDINGS OF THE 22nd ACM INTERNATIONAL CONFERENCE ON INFORMATION & KNOWLEDGE MANAGEMENT, 2013. Acm. p. 757-762.
- CRANDALL, David et al. **Mapping the World Photos**. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 2009. *Mapping the World's Photos*. Usa: Acm, 2009. p. 761-770.
- ELKI, Documentation. **ELKI Data Mining Framework**. Disponível em: <<http://elki.dbs.ifi.lmu.de/>>. Acesso em: 29 de Maio de 2015.
- ESTER, Martin et al. **A Density-Based Algorithm for Discovering Clusters**. In: PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 1996. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. Usa: AAAI, 1996. p. 226-231.
- HALL, Mark et al. **The Weka Data Mining Software: An Update**. In: SIGKDD EXPLORATIONS, 2009. *The Weka Data Mining Software: An Update*. Volume 11, Issue 1. 2009.
- HAYS, James. EFROS, Alexei A. **IM2GPS: Estimating Geographic Information from a Single Image**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2008. *IM2GPS: Estimating Geographic Information from a Single Image*. Usa: IEEE, 2008.
- HUANG, Anna. **Similarity Measures for text document Clustering**. In: PROCEEDINGS OF THE SIXTH NEW ZEALAND COMPUTER SCIENCE RESEARCH STUDENT CONFERENCE, 2008. *Similarity Measures for text document Clustering*. New Zealand, 2008. p. 49-56.
- JAIN, Anil K. MURTY, M. Narasimha. FLYNN, Patrick J. **Data clustering: a review**. In: ACM COMPUTING SURVEYS (CSUR), 1999. *Data Clustering: A Review*. Usa: Acm, 1999. v. 31, n. 3, p. 264-323.
- KASZNAR, Istvan Karoly; GONÇALVES, Bento Mario Lages. **Técnicas de Agrupamento Clustering**. In: INSTITUCIONAL BUSINESS CONSULTORIA INTERNACIONAL, 2009. *Técnicas de Agrupamento Clustering*. 2009.
- LACERDA, Yuri A. et al. **PhoCA: An extensible service-oriented tool for Photo Clustering Analysis**. In: 29th SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 2014. *PhoCA: An extensible service-oriented tool for Photo Clustering Analysis*. Curitiba: SBB, 2014. p. 227-232.
- MEYER, Andréia da Silva. **Comparação de Coeficiente de Similaridade usados em análise de agrupamento com dados marcadores moleculares dominantes**. 2002. 78 f. Dissertação (Mestrado) – Escola Superior de Agricultura Luiz de Queiroz. Universidade de São Paulo. Piracicaba, 2002.

OLIVEIRA, Tatyana Bitencourt Soares de. **Clusterização de Dados utilizando Técnicas de Redes Complexas e Computação Bioinspirada**. 2008. 112 f. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação. Universidade de São Paulo. São Carlos, 2008.

PELLEG, Dan. MOORE, Andrew W. **X-means: Extending K-means with Efficient Estimation of Number of Clusters**. 17TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 2000. *X-means: Extending K-means with Efficient Estimation of Number of Clusters*. Usa: ICML, 2000. (p. 727-734).

PHOTOMETADATA. **PhotoMetadata**. Disponível em: <<http://photometadata.org/>>. Acesso em: 28 de Maio de 2015.

RAPIDMINER. **RapidMiner Studio**. Disponível em: <<http://docs.rapidminer.com/studio/>> . Acesso em: 06 de Maio de 2015.

YANG, Yiyang. GONG, Zhiguo. U, Leong Hou. **Identifying Points of Interest by Self-Tuning Clustering**. In: PROCEEDINGS OF THE 34th INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 2011. *Identifying Points of Interest by Self-Tuning Clustering*. Usa: Acm, 2011. P 833-892.

## APÊNDICE A - SCRIPT UTILIZADO PARA *DOWNLOAD* DOS METADADOS DO SITE DE COMPARTILHAMENTO *FLICKR*

### **dados.sh**

```
#!/bin/bash
for pagina in $(seq 792) # Estrutura de repetição utilizada
com 792, que é o número de páginas de fotos do nordeste
disponíveis no Flickr
do
echo "$pagina"
LINK=https://api.flickr.com/services/rest/?method=flickr.photo
s.search&api_key=a391d133ae4d2411d8d04a6cde6f070f&bbox=-45%2C-
15%2C-33%2C0&extras=date-
taken%2Cgeo&per_page=500&page=$pagina&format=rest #Link com a
chave de 24h e os métodos disponíveis
wget $LINK

mv
"index.html?method=flickr.photos.search&api_key=a391d133ae4d24
11d8d04a6cde6f070f&bbox=-45%2C-15%2C-33%2C0&extras=date-
taken%2Cgeo&per_page=500&page=$pagina&format=rest" site`echo
$pagina`.txt
done
```

## APÊNDICE B - SCRIPT UTILIZADO PARA JUNTAR OS ARQUIVOS XML EM APENAS UM

### associar.sh

```
#!/bin/bash

# Criando o arquivo xml e juntando todos os xml do download em
# um só
touch definitivo.xml
echo "<?xml version='1.0' encoding='utf-8' ?>" >
definitivo.xml
echo "<rsp stat='ok'" >> definitivo.xml
echo "<photos page='1' pages='792' perpage='500'"
total="395601">" >> definitivo.xml

# adicionando os metadados dos arquivos separados para um
# único arquivo
for dados in $(seq 792)
do
    grep id site$dados.txt >> definitivo.xml
done

echo "</photos>" >> definitivo.xml
echo "</rsp>" >> definitivo.xml
```