UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC
CENTRO TECNOLÓGICO – CTC
CURSO DE SISTEMAS DE INFORMAÇÃO

ANÁLISE DE DADOS ESPAÇO-TEMPORAIS GERADOS POR DISPOSITIVOS MÓVEIS NA REDE SOCIAL TWITTER

RENATA DE JESUS SILVA

FLORIANÓPOLIS – SC

NOVEMBRO 2012

RENATA DE JESUS SILVA

ANÁLISE DE DADOS ESPAÇO-TEMPORAIS GERADOS POR DISPOSITIVOS MÓVEIS NA REDE SOCIAL TWITTER

Trabalho de conclusão de curso apresentado pela acadêmica Renata de Jesus Silva ao Curso de Sistemas de Informação da Universidade Federal de Santa Catarina, como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Luis Otavio Campos Alvares

UNIVERSIDADE FEDERAL DE SANTA CATARINA

FLORIANÓPOLIS – SC

NOVEMBRO 2012

RENATA DE JESUS SILVA

ANÁLISE DE DADOS ESPAÇO-TEMPORAIS GERADOS POR DISPOSITIVOS MÓVEIS NA REDE SOCIAL TWITTER

Trabalho de conclusão de curso apresentado pela acadêmica Renata de Jesus Silva ao Curso de Sistemas de Informação da Universidade Federal de Santa Catarina, como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Florianópolis, Novembro de 2012.

Orientador:

_____

Prof. Dr. Luis Otavio Campos Alvares

Banca Examinadora:

_____

Profª. Drª. Luciana de Oliveira Rech

_____

Prof. Dr. Ronaldo dos Santos Mello

AGRADECIMENTOS

Primeiro agradeço a Deus por tanta garra para ir atrás dos meus sonhos, força de vontade nos momentos em que cogitei desistir e determinação para traçar meu caminho com foco no que almejo.

Agradeço ao meu pai que, em praticamente toda a minha vida fez papel de pai e mãe. Meu sucesso é consequência da confiança dele em mim. Tenho também muito a agradecer às minhas irmãs Gisele e Juliana que sempre me motivaram, aconselharam e apoiaram. Dedico este trabalho a vocês, minha família! A vocês que sempre me incentivaram aos estudos e acreditaram na minha capacidade!

Agradeço muito ao meu namorado Victor à compreensão, apoio e presença mesmo nos momentos mais árduos; aos meus amigos, em especial o Marcelo que sempre esteve disposto a me ajudar nos momentos mais difíceis durante a graduação. Agradeço às meninas Janaína, Lara e Julia pela convivência, bons momentos e pela grande parceria em trabalhos, estudos e momentos de lazer. Também sou grata a todas as palavras sábias do Guedes que muito me guiou no início deste projeto.

Agradeço ao prof. Luis Otavio Alvares por me orientar neste trabalho, por toda a dedicação e por ter confiado no meu potencial. Aos professores da banca Luciana e Ronaldo por terem aceitado o convite, me sinto honrada por isto, pois ambos são exemplos de sabedoria e excelentes no que atuam.

Por fim, agradeço a todos os meus professores pelos ensinamentos cruciais na minha jornada! Eles contribuíram extremamente ao meu crescimento pessoal e à minha vida profissional.

RESUMO

Estamos vivenciando uma era em que o acesso à internet por meio de dispositivos móveis torna-se cada vez mais frequente. Por consequência, o *microblog* Twitter[1] tem registrado um aumento significativo de *tweets*[2] postados. Grande parte destas mensagens possui coordenada geográfica do local de onde foram emitidas e são acompanhados de informações temporais e, a partir disso, podem-se levantar informações relevantes que contribuem para o conhecimento do comportamento da população. Existem diversos estudos e ferramentas no mercado para análise e monitoramento sobre o que está acontecendo no Twitter. Todavia, há poucas pesquisas sobre análise com base nos dados georreferenciados nesta rede social. Torna-se, então, necessária a análise e extração de conhecimento, potencialmente útil, sobre estes dados espaço-temporais. O objetivo principal deste trabalho é utilizar técnica de mineração de dados e extrair conhecimento a partir de coleções de dados gerados por dispositivos móveis, originados por usuários que emitem mensagens na rede social Twitter. Em suma, este trabalho de pesquisa faz análises de dados espaço-temporais que permite conhecer melhor o comportamento dos usuários deste *microblog*. Para melhor visualização de dados na análise, foi realizada a implementação de geração do mapa com o resultado do algoritmo de mineração aplicado.

Palavras-chave: mineração de dados, dados espaço-temporais, *cluster*, DBSCAN.

1 https://twitter.com. Último acesso em novembro 2012.
2 Mensagens publicadas no Twitter são chamadas de *tweets*.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API	*Application Programming Interface*
GPS	*Global Position System*
GUI	*Graphical User Interface*
HTML	*Hyper Text Markup Language*
IBGE	Instituto Brasileiro de Geografia e Estatística
SGBD	Sistema Gerenciador de Banco de Dados
SIG	Sistema de Informação Geográfica
SQL	*Structured Query Language*
SPAM	*Stupid, Pointless Annoying Message*
UFSC	Universidade Federal de Santa Catarina

SUMÁRIO

1. INTRODUÇÃO

Com a popularização das redes sociais, o acesso à informação tornou-se mais ágil assim como mensagens publicadas pelas mesmas. O volume de dados gerado pelo Twitter – que é uma rede social baseada em mensagens instantâneas de até 140 caracteres – originados de dispositivos móveis, torna viável o estudo sobre o comportamento, em relação ao tempo e localização, dos usuários desta rede. Analisar milhões de dados, publicados diariamente no Twitter, é muito trabalhoso e inviável manualmente. Uma alternativa para este problema é aplicar técnicas de mineração de dados, o que favorece o desempenho do processo de descoberta de conhecimento e propicia melhor eficiência computacional. Existem várias pesquisas realizadas na área de análise de dados espaço-temporais. Todavia, o estudo baseado nos dados georreferenciados do Twitter é pouco explorado, mesmo porque o recurso de geolocalização do *microblog* é recente, iniciou-se no ano de 2010. Com base nisso, o trabalho proposto tem o foco em mineração de dados utilizando a base de dados do Twitter, com *tweets* que possuem coordenadas geográficas.

Pode-se afirmar que trabalhar com dados espaço-temporais em rede social é uma tarefa desafiadora, pois existe um fluxo muito grande de dados além de estes dados serem complexos. A carência de pesquisas nesta área, considerando inclusive o foco na cidade de Florianópolis, indica que este nicho é interessante. Para trabalhar com dados espaço-temporais, existem bancos de dados espaciais que dão suporte a utilização de operações e relacionamentos espaciais.

Um aspecto determinante, e também motivador, para o desenvolvimento deste trabalho é que o número de pessoas que usam dispositivos móveis é crescente. Isto porque o acesso a estes está facilitado devido à queda de custos e várias opções para todo perfil de usuário. Outro ponto relevante é que o número de usuários de redes sociais é grande e tende a aumentar e, por consequência, a massa de dados gerada por estas também é crescente. A rede social Twiiter, segundo o blog oficial do mesmo, totaliza cerca de 140 milhões de usuários e cerca de 340 milhões de *tweets* publicados por dia.

Com este trabalho, a partir de aplicação de técnica de mineração de dados, espera-se analisar e extrair conhecimento de dados espaço-temporais originados de usuários do Twitter. Para que seja extraído conhecimento desta base, foi utilizada técnica de agrupamento.

O objetivo geral deste trabalho é, utilizando ferramentas disponíveis na computação, analisar uma massa de dados georreferenciados, gerada pela rede social Twitter, a fim de

obter informações potencialmente úteis, extrair conhecimento detectando tendências, padrões ou novos conhecimentos. O conjunto de dados para o trabalho limitou-se à cidade de Florianópolis. Por se tratar de um estudo acadêmico, a quantidade de dados utilizada foi suficiente para o encontro de resultados plausíveis.

Os objetivos específicos são análises gerais dos *tweets* que possuem localizações geográficas. São exemplos destas análises:

- Identificação de regiões densas (pontos de interesse) nos bairros de Florianópolis, isto é, detectar locais com alto número de registro de *tweets*;
- Frequência por local. Detectar primeiro estas informações, com análises exploratórias, para identificar a quantidade de dados que cada bairro possui e o que, estatisticamente, esta massa representa em todo o conjunto de dados;
- Formação de agrupamentos – os grupos são formados em regiões que possuem grande quantidade de *tweets* – nos bairros de Florianópolis em diferentes turnos do dia (manhã, tarde e noite) e dia da semana. Com isto, será possível analisar o comportamento dos usuários do Twitter em um determinado bairro de Florianópolis em diferentes situações;
- Comparação entre bairros de Florianópolis para analisar o comportamento dos usuários do Twitter com relação ao tempo e localização;
- Detecção de palavras mais utilizadas em grupos encontrados com o algoritmo de densidade DBSCAN;
- Acrescentar funcionalidades à ferramenta Weka[3] para melhorar a capacidade de resposta à pesquisa, incluindo a geração de mapa para visualização de *clusters*.

3 http://www.cs.waikato.ac.nz/ml/weka. Site oficial do Weka. Último acesso em novembro 2012.

2. CONCEITOS BÁSICOS

Para que este trabalho seja bem compreendido, esta seção apresenta os principais conceitos, técnicas e ferramentas utilizadas no desenvolvimento do trabalho.

2.1 REDE SOCIAL TWITTER E RECURSO DE GEOLOCALIZAÇÃO

Redes sociais permitem que usuários compartilhem informações rapidamente e, estas informações podem ser amplamente dispersadas na rede. Isto as torna uma importante plataforma de disseminação de informação, descoberta de conteúdo e compartilhamento de informação (BHAT *et al*, 2011). Este autor também aborda o serviço de localização do Twitter, o qual permite que usuários de dispositivos móveis disponibilizem sua localização geográfica no momento em que publica seu *tweet*.

Para que o Twitter consiga identificar as coordenadas geográficas das mensagens publicadas, estas devem ser originadas de dispositivos móveis que possuam *Global Position System* (GPS). A posição geográfica do dispositivo é calculada por satélite, mas caso ocorra perda de sinal a geolocalização é realizada utilizando triangulação. Esta perda de sinal geralmente acontece quando o usuário encontra-se em locais fechados como *shopping centers*, por exemplo.

2.2 DADOS ESPAÇO-TEMPORAIS

Dados espaciais, ou geográficos, possuem informação de localização e representação do objeto geográfico. Os dados são representados por um tipo geométrico, como polígonos, linhas, pontos e outros tipos de geometrias complexas. Aparelho GPS, que são comumente usados em veículos, é exemplo de aplicativo que gera dado espacial. Pode-se destacar que estes são capazes de definir uma rota de uma região para outra e apontar locais de interesse mais próximos da localização do motorista.

Seguindo com o exemplo do GPS, é interessante atribuir informação temporal aos dados. Dados espaço-temporais possuem também a caratcterística temporal. Isto sugere que este tipo de dado se move com o tempo.

A base de dados para este estudo foi disponibilizada com três características interessantes no ponto de vista espaço-temporal: latitude, longitude e data da publicação dos

tweets. Dados espaciais necessitam operações complexas que não são disponíveis em Sistema Gerenciador de Banco de Dados (SGBDs) convencionais. Portanto, existem SGBDs com extensão à Sistema de Informação Geográfica (SIG) que armazenam e manipulam estes dados. Para que o banco de dados geográfico reconheça e trabalhe com os dados espaciais, estes dados precisam ser representados com tipos geométricos. Assim, atributos de latitude e longitude – que eram apenas atributos do tipo "*double*" – foram convertidos para um ponto, que é de tipo "*geometry*". A criação da coluna geométrica e conversão dos atributos latitude e longitude está no Anexo A – DML 1.

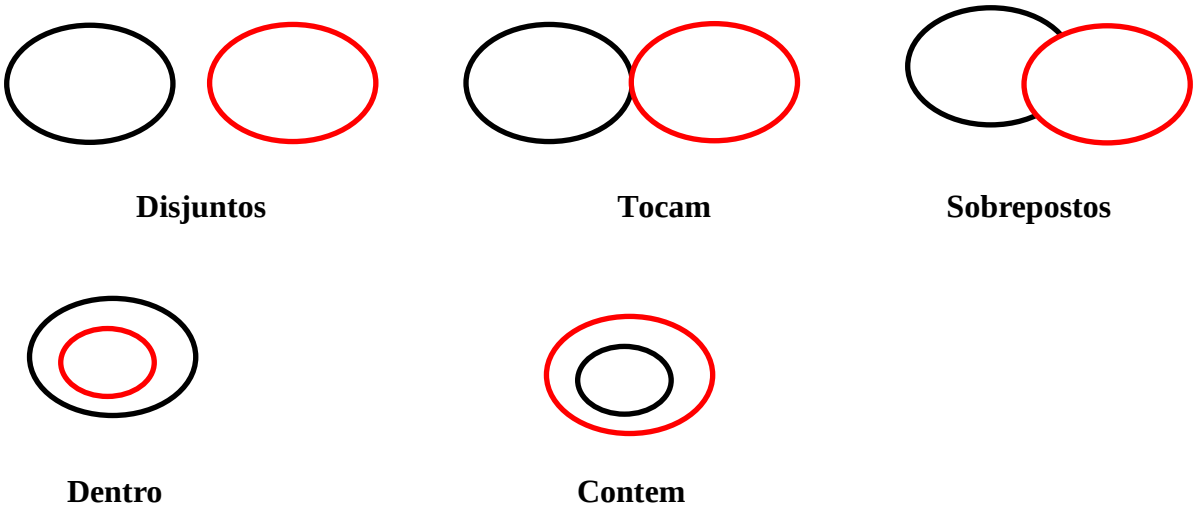
2.2.1 Tipos de Representações Espaciais

É crucial o conhecimento de relações e representações de geometrias espaciais na análise de dados de SIG. Segundo Egenhofer (1991), a topologia do modelo espacial de dados é definida em diferentes dimensões:

- 0-dimensional: ponto ou nó. É um objeto com a mínima dimensão, por exemplo, um ***tweet***.
- 1-dimensional: linha. Ligação de um ponto a outro, por exemplo, o **percurso ou trajetória** que um usuário faz durante o dia.
- 2-dimensional: área ou polígono. Sequencias contendo três ou mais pontos que não se intersectam. Por exemplo, **bairro**.
- Tridimensional: superfície. Por exemplo, uma construção.

2.2.2 Tipos de Relacionamentos Espaciais

Objetos espaciais se relacionam de acordo com a distância, ordem ou topologia. Distância nada mais é que a medida entre dois objetos. Ordem indica a referencia de um objeto a outro conforme a posição no espaço (norte, sul, leste, oeste). Já os principais relacionamentos, conforme a topologia, são ilustrados a seguir.



Disjuntos **Tocam** **Sobrepostos**

Dentro **Contem**

Além destas topologias representadas acima, existem outros dois relacionamentos relevantes, que são melhores descritos desta maneira: (i) **Cruzam (*crosses*)**: um objeto atravessa o outro; (ii) **Igual**: dois objetos são idênticos.

2.3 MINERAÇÃO DE DADOS E DESCOBERTA DE CONHECIMENTO

HAN (2006) sintetiza mineração de dados como extração, ou mineração, de conhecimento em uma grande quantidade de dados. Analogamente, é como encontrar um pequeno conjunto de pepita de ouro em uma grande quantidade de matéria prima. Mineração de dados pode ser vista como uma etapa no processo de descoberta de conhecimento. Este autor descreve as etapas de data mining da seguinte forma: limpeza e integração dos dados, seleção e transformação dos dados, mineração de dados, avaliação de padrões e, por fim, apresentação do conhecimento.

Tarefas de mineração de dados podem ser classificadas em duas categorias: (i) **Descrição** que se caracteriza por encontrar padrões que descrevem os dados; (ii) **Predição** que prediz um valor desconhecido de variável por meio de valores conhecidos de outras variáves.

Segundo HAN (2006), há casos que não se sabe que tipos de padrões podem ser encontrados com a mineração de dados. É importante utilizar um sistema que permite aplicar diversas tarefas de mineração de dados. Assim, podem ser aplicadas tarefas de mineração de dados e detectar padrões em diferentes granularidades.

As principais tarefas consitem em: Classificação, Associação, Agrupamento, Regressão e Detecção de desvios. A descrição de algumas destas tarefas é mostrada a seguir, baseando-se em HAN (2006).

2.3.1 Classificação

Classificação é uma tarefa preditiva que busca um modelo que distingue as classes dos dados que não possuem valor definido. O objetivo da classificação é, baseando-se em um grupo de objetos classificados, detectar a qual classe um objeto pertence. A partir de um conjunto de registros com valores conhecidos (chamado de conjunto de treinamento), onde cada registro possui atributos e um destes é a classe, é aplicado um modelo para classificar os registros que contem dados faltantes. A classificação de objetos pode ser realizada por árvores de decisão, redes neurais, redes bayesianas, máquinas de vetores de suporte e regras de decisão, entre outros métodos.

2.3.2 Associação

Associação é uma tarefa que determina dependência ou correlações entre dados. Esta tarefa identifica padrões frequentes que aparecem em um conjunto de registros. Em um conjunto de transações, regra de associação é determinada na predição de ocorrência de itens baseado na ocorrência de outros itens na transação. A regra de associação é: X -> Y, tal que X e Y são conjuntos disjuntos de atributos e ocorrem juntos em uma transação. Uma regra de associação é desinteressante caso não satisfaça duas condições: (i) **Suporte mínimo** que é a percentagem das transações que contem os atributos X e Y; (ii) **Confiança** que estima a probabilidade que Y aparece nas transações que contem X.

Um exemplo prático de regra de associação é, {local: estádio de futebol} -> {palavra frequente: juiz}, isto indica que quem está em estádio de futebol comenta sobre juiz.

2.3.3 Agrupamento (*Clustering*)

Agrupamento é uma técnica que forma grupos de dados de modo a maximizar a similiaridade dos dados que pertencem a um mesmo grupo (*cluster*) e minimizar a similaridade entre grupos distintos. Isto é, *clusters* são formados por objetos que possuem grande similaridade entre dados do mesmo *cluster*, contudo são diferentes de objetos contidos em outros *clusters*.

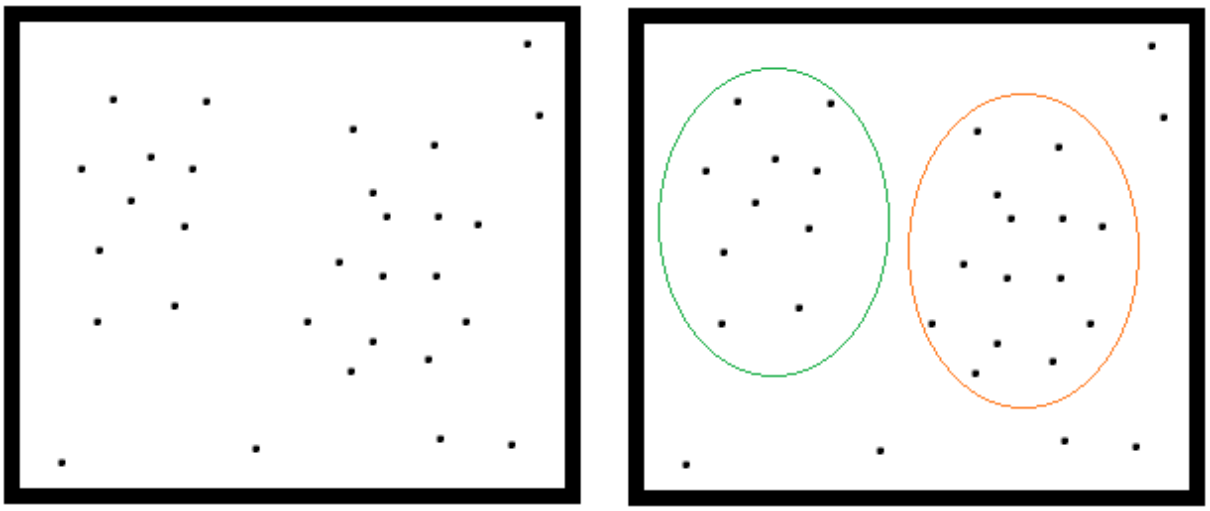


Figura 1 - Pontos iniciais (esquerda) e pontos agrupados (direita).

A Figura 1 apresenta um exemplo de agrupamento dos dados. No lado esquerdo da figura são representados os pontos iniciais da base de dados e, no lado direito é representada uma maneira de agrupar tais pontos. Dependendo do algoritmo de agrupamento aplicado, podem ser formados diferentes grupos. A quantidade de grupos e a quantidade de objetos em cada grupo dependem dos parâmetros passados ao algoritmo de *cluster* escolhido.

3. A FERRAMENTA WEKA

Para mineração e análise de dados, foi utilizado o software Weka. Weka é uma ferramenta criada pela Universidade de Waikato, situada na Nova Zelândia. Este software é de código aberto, utiliza *GNU General Public License* e foi desenvolvido na linguagem de programação Java. Para visualização e interação com usuário, contem uma *Graphical User Interface* (GUI).

Figura 2 - Tela inicial do software Weka.

De acordo com o *site* oficial do *software* Weka, este é uma coleção de algoritmos para execução das tarefas de mineração de dados. Os algoritmos podem ser aplicados diretamente de um conjunto de dados (banco de dados, planilha eletrônica, etc.) ou ser utilizados em uma aplicação Java – importando as bibliotecas necessárias. Weka contém ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização.

A ferramenta de mineração de dados Weka foi escolhida para este projeto por permitir o uso do algoritmo de *clustering* (agrupamento) DBSCAN – também é chamado de algoritmo baseado em densidade, pois forma *clusters* baseando-se na distribuição dos dados. Para ser considerada uma região densa, deve haver um número mínimo de pontos dentro de um raio (épsilon). Um dos objetivos propostos é identificar pontos de interesse. Por isso foi escolhido este algoritmo para agrupamento de dados.

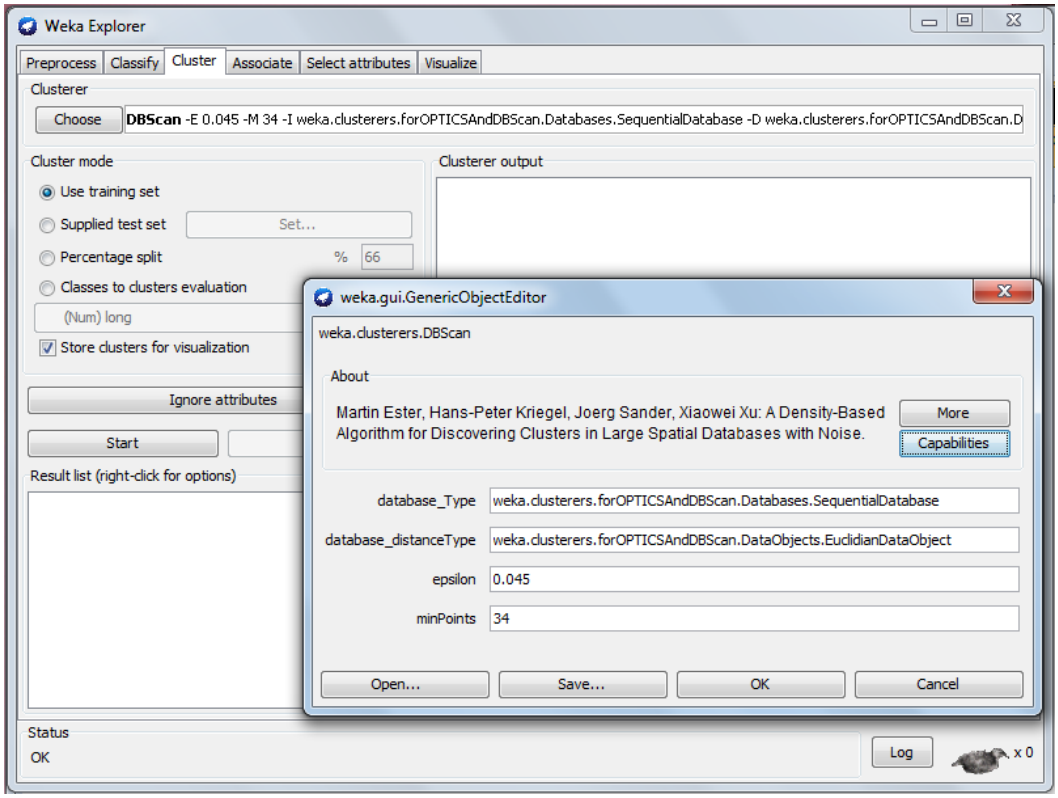


Figura 3 - Algoritmo DBSCAN no software Weka.

Segundo BAJPAI (2012), as vantagens do algoritmo DBSCAN são: (i) não requer que o usuário saiba previamente o número de *clusters* em um conjunto de dados; (ii) é capaz de encontrar arbitrariamente *clusters* em diferentes formatos; (iii) é tolerante a ruídos (se não for respeitado o número mínimo de pontos, este algoritmo considera o ponto como *noise* (ruído)); (iv) requer apenas dois parâmetros (mínimo de pontos e épsilon).

Por conseguinte, as desvantagens deste algoritmo são: (i) é difícil encontrar um valor para épsilon (tal que épsilon é o raio de um círculo), pois se o valor deste não for adequado não se terá um bom resultado; (ii) não permite gerar grupos em dados que possuem diferentes densidades, caso a combinação dos parâmetros mínimo de pontos e épsilon não for escolhida apropriadamente. Por exemplo, havendo um total de 500 registros, sendo que o número mínimo de pontos definido é 150, se existir somente uma área densa (considerando o valor do épsilon) com 200 pontos e o restante (300 pontos) estiverem completamente espalhados, então haverá somente um *cluster* (com 200 registros) e o restante será definido como ruído.

É importante ressaltar que o algoritmo DBSCAN da ferramenta Weka não manipula dados do tipo geográficos. Todavia, se os campos passados para este algoritimo de *cluster* forem os campos “latitude” e “longitude” (a base de dados do Twitter possui estes dados) o

Weka irá agrupar os pontos próximos entre si já que estes são similares. Outro detalhe a ser considerado é que no visualizador de *clusters* formados da ferramenta Weka é possível observar os pontos dos *clusters* plotados pelas coordenadas X e Y escolhidas pelo usuário. Neste visualizador do Weka consegue-se apenas visualizar a distância dos *clusters* e sua distribuição. Contudo não há nenhuma informação geográfica como ruas, casas, *shopping centers*, restaurantes, rios, montanhas, etc. Assim houve a necessidade de adaptar o *software* Weka para que este pudesse gerar um mapa geográfico.

4. DESENVOLVIMENTO DO PROJETO

Este trabalho foi realizado utilizando dados do Twitter e aplicando métodos de mineração de dados sobre eles.

A base de dados utilizada para esta pesquisa possui apenas uma amostragem dos dados gerados no *microblog*. Os dados contidos nesta amostragem limitam-se aos *tweets* publicados em Florianópolis – capital do estado de Santa Catarina.

Para que sejam aplicadas funções espaciais sobre dados espaço-temporais, é necessário um banco de dados geográfico, já que os bancos de dados convencionais são limitados em relação a consultas que envolvem tipos de dados geométricos e funções que exigem cálculos complexos. Por esta razão, para a manipulação destes dados, será utilizado a extensão PostGIS[4] do PostgreSQL[5]. A escolha por este SGBD deu-se por ser uma ferramenta gratuita, de código fonte aberto e, certamente, por permitir o uso de objetos de SIG.

4.1 CONSIDERAÇÕES SOBRE A BASE DE DADOS DO IBGE

Para que fosse possível alocar os *tweets* nos bairros de Florianópolis, foi utilizada a base de dados extraída do *site* do Instituto Brasileiro de Geografia e Estatística (IBGE). O IBGE disponibiliza arquivos *shapefile* contendo informação geográfica das cidades e estados do Brasil (IBGE, 2012).

Para visualização destes dados, foi utilizada a ferramenta Quantum GIS[6] (versão 1.7.1). Já para análise e divisão da tabela dos dados do IBGE, foi utilizado a extensão PostGIS citada anteriormente.

Uma das dificuldades encontradas no decorrer do projeto foi que o IBGE não disponibilizou o arquivo *shapefile* contendo a separação de bairros de Florianópolis, somente foram disponibilizados arquivos com os setores censitários. Desta maneira, um bairro possui vários setores censitários, porém a junção de todos estes setores pode não totalizar um bairro (pois há locais como morros, dunas, etc. que não pertencem a setor nenhum). A partir disso foram feitas análises exploratórias sobre os dados da tabela de Florianópolis gerada e não foi detectado padrão algum sobre o nome do bairro. Isto por que, como a tabela, inicialmente, tratava-se de setores censitários, o nome do bairro indicado referenciava o setor censitário

4 http://postgis.refractions.net. Último acesso em novembro 2012.
5 http://www.postgresql.org. Último acesso em novembro 2012.
6 http://www.qgis.org. Último acesso em novembro 2012.

propriamente dito. Assim, existem na base de dados cerca de noventa bairros distintos, mas de acordo com o *site* da Prefeitura Municipal de Florianópolis[7] (PMF), existem 35 bairros situados na cidade. Estes bairros estão listados na Tabela 1.

Abraão	Coqueiros	Monte Verde
Agronômica	Córrego Grande	Pantanal
Balneário	Costeira do Pirajubaé	Pântano do Sul
Barra da Lagoa	Estreito	Ratones
Bom Abrigo	Ingleses do Rio Vermelho	Ribeirão da Ilha
Cachoeira do Bom Jesus	Itacorubi	Saco dos Limões
Campeche	Itaguaçu	Saco Grande
Canasvieiras	Jardim Atlântico	Santa Mônica
Canto	João Paulo	Santo Antônio de Lisboa
Capoeiras	José Mendes	São João do Rio Vermelho
Centro	Lagoa da Conceição	Trindade
Coloninha	Monte Cristo	

Tabela 1 - Lista de bairros do munícipio de Florianópolis, segundo a PMF.

A Figura 4 mostra o mapa disponibilizado pelo IBGE e, ao lado direito desta figura, foi dado *zoom* na ilha de Florianópolis. A ferramenta Quantum GIS permite que sejam visualizados dados provindos do arquivo *shapefile* e também suporta conexão com o SGBD geográfico.

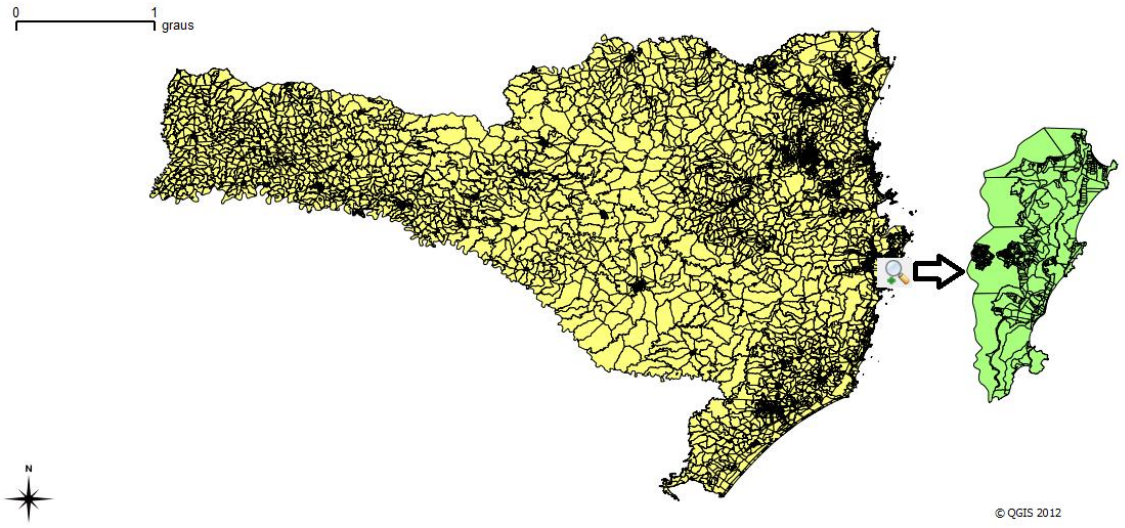


Figura 4 - Setores censitários de Santa Catarina visualizado pela ferramenta Quantum GIS.

7 http://www.pmf.sc.gov.br/. Último acesso em novembro 2012.

Como o nível de granularidade a ser trabalhado primeiramente é bairro, foi utilizado a função ST_UNION do PostGIS (Anexo A – DDL 1), que faz a união de setores censitários, para que formem os bairros. Assim, o mapa formado com a separação de bairros foi visualizado na ferramenta Quantum GIS conforme a Figura 5.

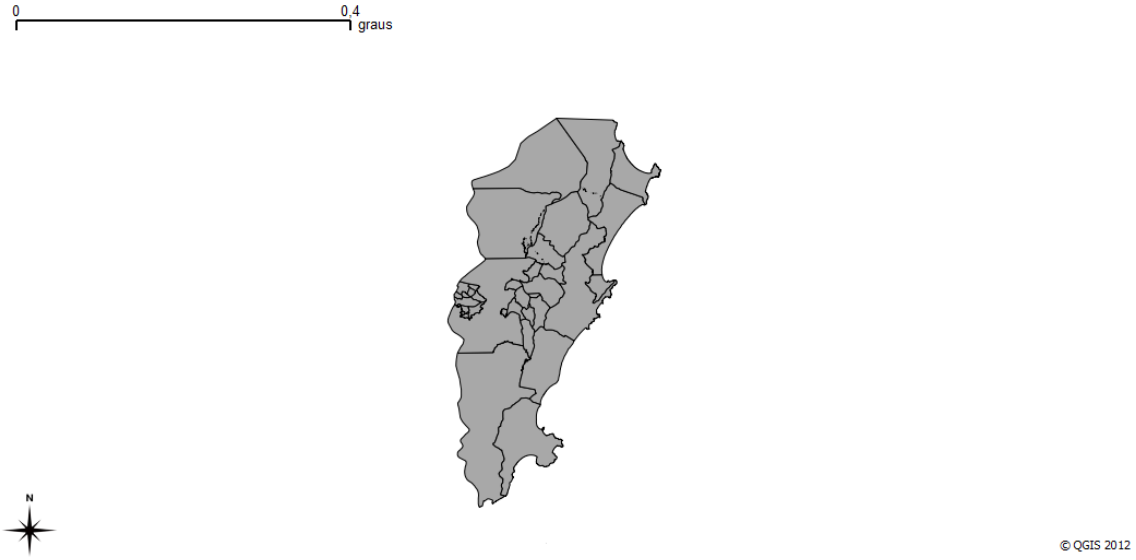


Figura 5 - Mapa de Florianópolis visualizado pela ferramenta Quantum GIS.

4.2 CONSIDERAÇÕES SOBRE A BASE DE DADOS DO TWITTER

Previamente, os dados do Twitter foram coletados e preparados em um trabalho de pesquisa de autoria de Augusto Dias Pereira dos Santos – estudante do Programa de Pós-Graduação em Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul.

Duas tabelas foram criadas para inserir os dados coletados no trabalho supracitado, vide *script* no Anexo A – DDL 2. A partir das tabelas criadas, foram inseridos os dados no banco pelo arquivo *Structured Query Language* (SQL) disponibilizado. No total, foram cerca de 19 milhões de registros inseridos na base. Estes registros foram publicados no Twitter entre os meses de abril e novembro de 2011. Destes registros, foram selecionados apenas os *tweets* de Florianópolis – totalizando 172.838 dados.

Após a criação e carga no banco de dados, decidiu-se adicionar a coluna bairro, para adequar os dados a este trabalho de pesquisa. Então, a tabela contendo os bairros e a tabela do Twitter foram cruzadas, utilizando a função ST_CONTAINS do PostGIS, com o objetivo de popular a coluna "bairro" pertencente à tabela de *tweets* de Florianópolis. Antes de atualizar a tabela de *tweets*, foi adicionado um índice espacial geométrico que, em teste, mostrou

aumentar cerca de três vezes a velocidade da consulta. A *query* utilizada para atualização pode ser observada no Anexo A – DML 2.

Ao sobrepor os dados da amostragem de *tweets* na tabela de bairros, com o suporte da ferramenta Quantum GIS, foram verificados que alguns dados estavam fora do limite da cidade em estudo.

A Figura 6 mostra este problema, onde os pontos azuis são os *tweets*.

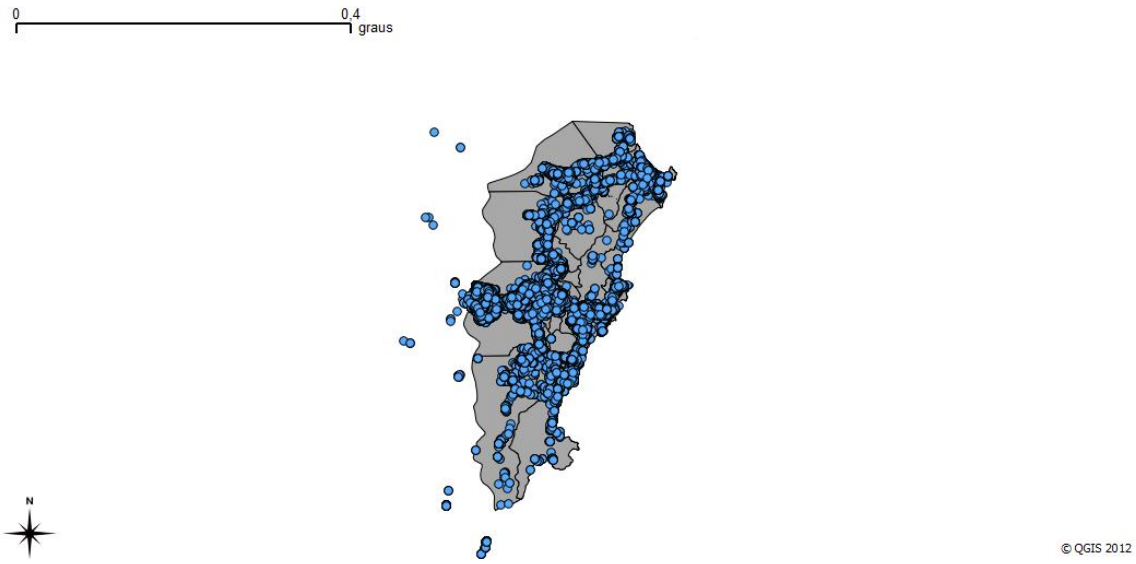


Figura 6 - Mapa de Florianópolis com *tweets* visualizado pela ferramenta Quantum GIS.

Com a Figura 6, pode-se perceber que alguns dos dados não pertencem à Florianópolis, mesmo que a cidade indicada pela tabela do Twitter era esta. Fazendo análises exploratórias destes dados, descobriu-se que, de fato, o nome do bairro na base de dados estava nulo. Além disso, por meio da ferramenta Google Maps[8], foi verificado que os pontos latitude e longitude realmente não pertenciam aos bairros de Florianópolis.

Os dados errôneos, pesquisando nas informações disponibilizadas no *site* do IBGE e verificação no Google Maps, pertencem a cidades vizinhas de Florianópolis e, esporadicamente, uns podem estar localizados no mar (tal que, provavelmente, ocorreu uma imprecisão com margem de erro maior que o esperado. Também há possibilidade de que o *tweet* foi postado realmente no mar, por exemplo, em um barco). A quantidade destes dados é

8 https://maps.google.com.br/. Último acesso em novembro 2012.

de 426 registros e representam, aproximadamente, 0,25% da quantidade de dados da amostragem de Florianópolis.

Os dados foram explorados com o intuito de conhecer quanto cada bairro representa no total de registros estudados. Ressaltando, a massa de dados na base de *tweets* representa um total de 18.950.286 registros. Além disso, é importante destacar que a quantidade de usuários distintos, na base de usuários, é de 211.898.

Definiu-se que o foco de estudo inicial é a capital do estado de Santa Catarina: Florianópolis. Para tanto, foram levantadas algumas informações relevantes a partir da base de dados utilizada para este estudo. Algumas delas são:

- Na base de dados de estudo, o estado de Santa Catarina possui 664.857 registros, isto é, 3,5% do total dos dados. A quantidade de usuários distintos neste estado é de 10.456, o que representa quase 5% do total de usuários do Brasil.
- A cidade de Florianópolis possui 172.838 registros. Isto representa quase 26% do total de dados de Santa Catarina.
- Usuários que tuitaram 10 ou mais vezes em Florianópolis, representam 95% do total de registros total desta cidade.
- O bairro onde mais *tweets* foram emitidos foi o Centro, seguido de Trindade e Itacorubi.

Na Tabela 2 é mostrada uma visão geral dos dados trabalhados (para isto, foram desconsiderados registros de usuários que tuitaram menos de 10 vezes e registros sem informação de bairro. Isto será detalhado posteriormente). O total de usuários distintos mostrado nesta tabela é referente à quantidade de usuários diferentes que publicaram *tweets* em Florianópolis. Não significa que é a soma de todos os usuários distintos por bairro, pois o mesmo usuário pode ter tuitado de vários bairros.

	Tweets	Usuários Distintos
Abraão	1621	86
Agronômica	3061	393
Balneário	1815	104
Barra da Lagoa	517	88
Bom Abrigo	252	43
Cachoeira do Bom Jesus	1576	151
Campeche	4992	198
Canasvieiras	7545	344
Canto	4837	207
Capoeiras	4434	314
Centro	45552	1108
Coloninha	599	56
Coqueiros	6879	246
Córrego Grande	4673	287
Costeira do Pirajubaé	333	67
Estreito	4077	212
Ingleses do Rio Vermelho	5771	210
Itacorubi	13462	427
Itaguaçu	350	72
Jardim Atlântico	1455	135
João Paulo	2571	142
José Mendes	76	42
Lagoa da Conceição	4654	425
Monte Cristo	495	31
Monte Verde	2253	247
Pantanal	1457	131
Pântano do Sul	579	45
Ratones	105	31
Ribeirão da Ilha	7393	544
Saco dos Limões	2158	130
Saco Grande	4472	263
Santa Mônica	4266	454
Santo Antônio de Lisboa	2957	313
São João do Rio Vermelho	527	42
Trindade	16545	530
Total	**164309**	**1432**

Tabela 2- Análise exploratória dos dados.

A Tabela 2 retrata a quantidade de dados disponíveis na base, assim como a quantidade de usuários distintos que tuitaram em cada bairro. Em destaque, o bairro com maior número de *tweets* foi o Centro, com 41,1% do total de registros. O bairro com a maior

quantidade de usuários distintos também foi o centro. Em contrapartida, o bairro com menos *tweets* foi José Mendes.

4.3 DESENVOLVIMENTO JAVA: ALTERAÇÕES NA FERRAMENTA WEKA

Antes de minerar os dados deste estudo, foram implementadas novas funcionalidades no Weka. Este desenvolvimento foi feito na linguagem de programação Java.

A necessidade da alteração deste código deu-se pela falta de informação na visualização dos dados minerados pelo algoritmo DBSCAN. Antes do desenvolvimento, o resultado gerado era visualizado conforme a Figura 7.

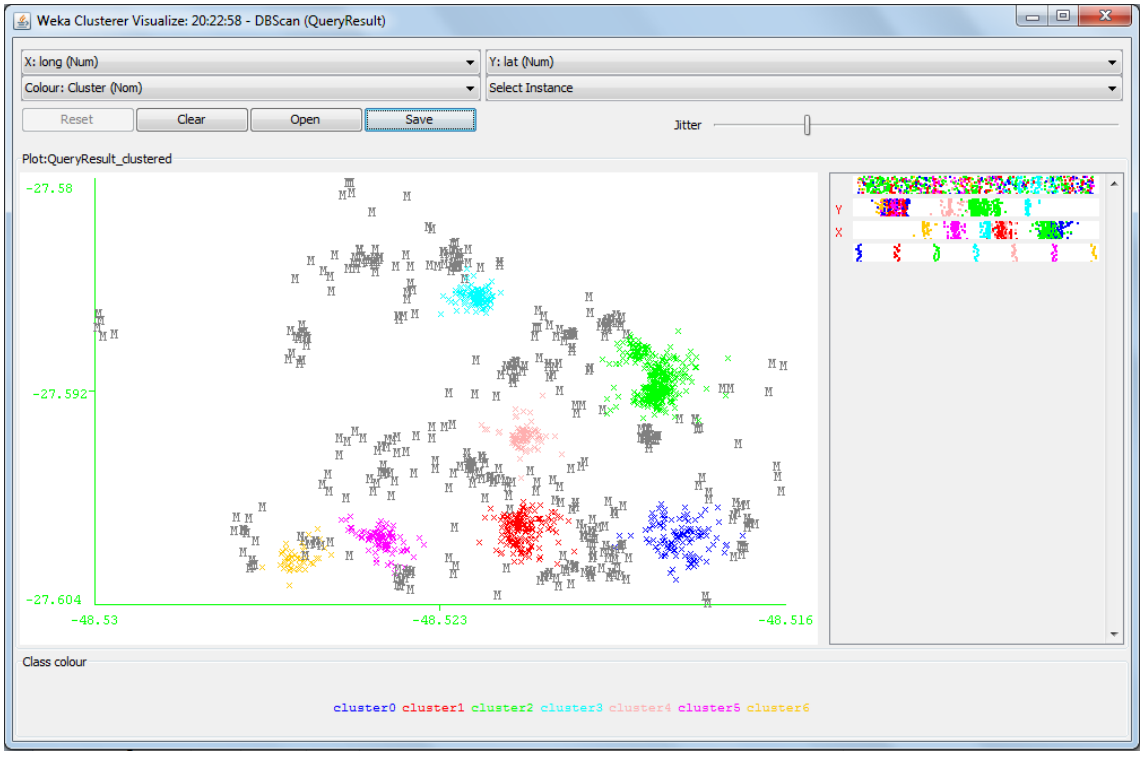


Figura 7 - Resultado dos *clusters* formados visualizado pela ferramenta Weka.

Estes *clusters* foram formados pelos registros do bairro Trindade, período da manhã e dia de semana. Os parâmetros utilizados foram: (i) épsilon = 0,045; (ii) mínimo de pontos = 34; (iii) tipo de distância = Euclidiana.

Com esta imagem, é possível perceber que a análise de dados não seria eficaz, pois não é possível identificar o que há na região destes pontos. A maneira da disposição dos

pontos também é confusa, pois não tem como saber exatamente qual é o ponto central dos grupos formados.

A partir desta falta de informações, a qual é crucial para este estudo, decidiu-se por acrescentar métodos e classes no código da ferramenta Weka. A finalidade desta implementação foi adaptar o código desta ferramenta de maneira que se chegue ao objetivo do estudo com mais facilidade, agilidade e evitando esforço humano. Assim, tornou-se mais eficiente a capacidade de extração de conhecimento a partir das análises realizadas.

Para a visualização dos *clusters*, foram considerados 4 aspectos: (i) a ferramenta Weka foi adaptada para gerar um mapa onde cada marcador representa o centróide de um *cluster*; (ii) os pontos plotados no mapa são somente os centróides de cada *cluster*; (iii) geração de um único mapa contendo todos os centróides de todas as consultas executadas no Weka; (iv) ao clicar em um marcador, podem ser visualizadas as 15 palavras mais frequentes do *cluster*.

As principais modificações e classes adicionadas são apresentadas a seguir.

4.3.1 DBSCAN

As modificações feitas no algoritmo DBSCAN do *software* Weka foram:

- Cálculo dos centróides de cada *cluster* encontrado. Este dado a mais se tornou necessário para que, posteriormente, o usuário consiga visualizar no mapa somente um ponto (o centróide) entre os pontos de um *cluster*.
- Cálculo automatizado do parâmetro "mínimo de pontos", onde, para todas as consultas SQL, este parâmetro passado foi 2,5% da quantidade de dados que esta consulta retorna. Uma exceção é que o mínimo de pontos será sempre 12, caso o resultado deste cálculo seja menor que isto.
- Na geração dos *clusters* formados pelo DBSCAN, ocorrem inserções no banco de dados para o armazenamento do número do *cluster* ao qual cada *tweet* pertence.

4.3.2 Google Maps

O mapa utilizado no projeto é fornecido pela *Application Programming Interface* (API) do Google Maps. Para possibilitar a inserção de múltiplos pontos com ícones personalizados foi adaptado o *script* do *site* Link Nacional. Com este script é possível indicar latitude, longitude, ícone e descrição para cada ponto.

Foi criada uma classe responsável por gerar um mapa de extensão HTML. Nele são mostrados os centróides dos *clusters* e, além disso, as 15 palavras mais frequentes de cada *cluster* podem ser visualizadas com o clique de mouse sobre o marcador desejado. Para abrir o mapa formado pelo Weka, o usuário deve utilizar o navegador web de preferência.

4.3.3 Criação de Tabelas

Foram criadas duas tabelas no banco de dados (Anexo A – DDL 3):

1) "cluster_dbscan": Tabela informativa com o número da consulta e descrição da consulta. Por exemplo, número da consulta = 1; descrição da consulta = Trindade, manhã, final de semana.
2) "rel_cluster_tweet": Tabela criada com objetivo de armazenar o número do *cluster* e indicar qual a consulta e *tweet* que o mesmo referencia.

4.3.4 Resultado da Implementação

Após a implementação realizada, os *clusters* podem ser visualizados como na Figura 8. Para geração desta figura, foram utilizados os mesmos dados e parâmetros da Figura 7. É importante ressaltar que a Figura 8 é um exemplo do mapa gerado, desenvolvido neste projeto.

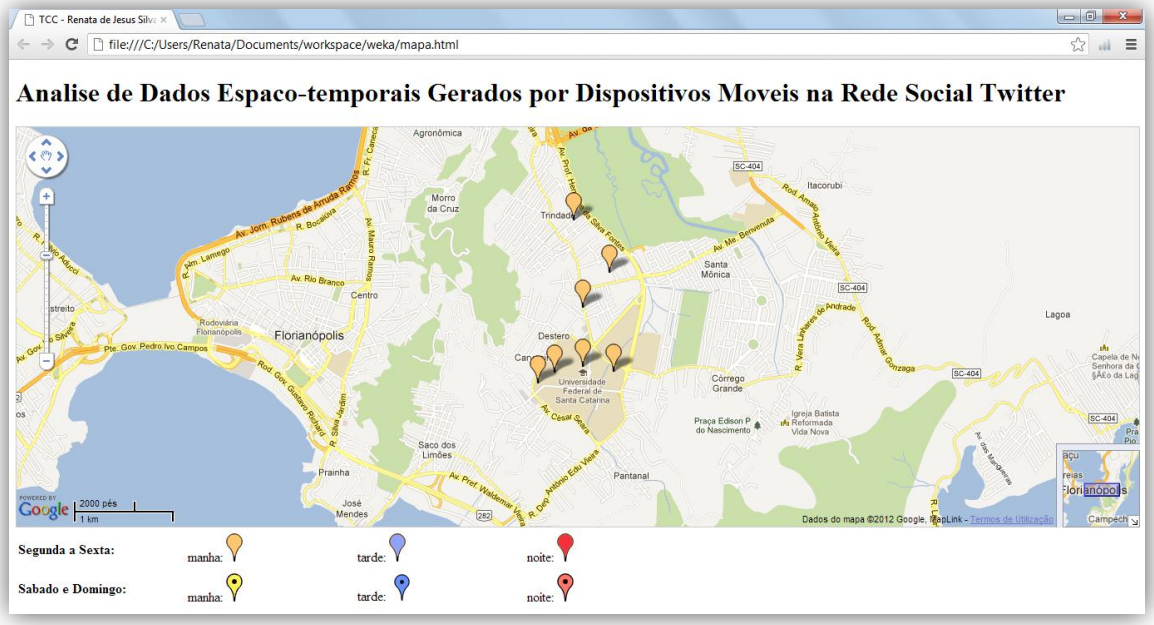


Figura 8 – Visualização do mapa com os centróides dos *clusters* – bairro Trindade.

Para melhor compreesão, o mapa é composto de uma legenda localizada no rodapé que referencia a cor utilizada para cada consulta. Na Figura 9, são mostradas as palavras mais frequentes de um *cluster*.

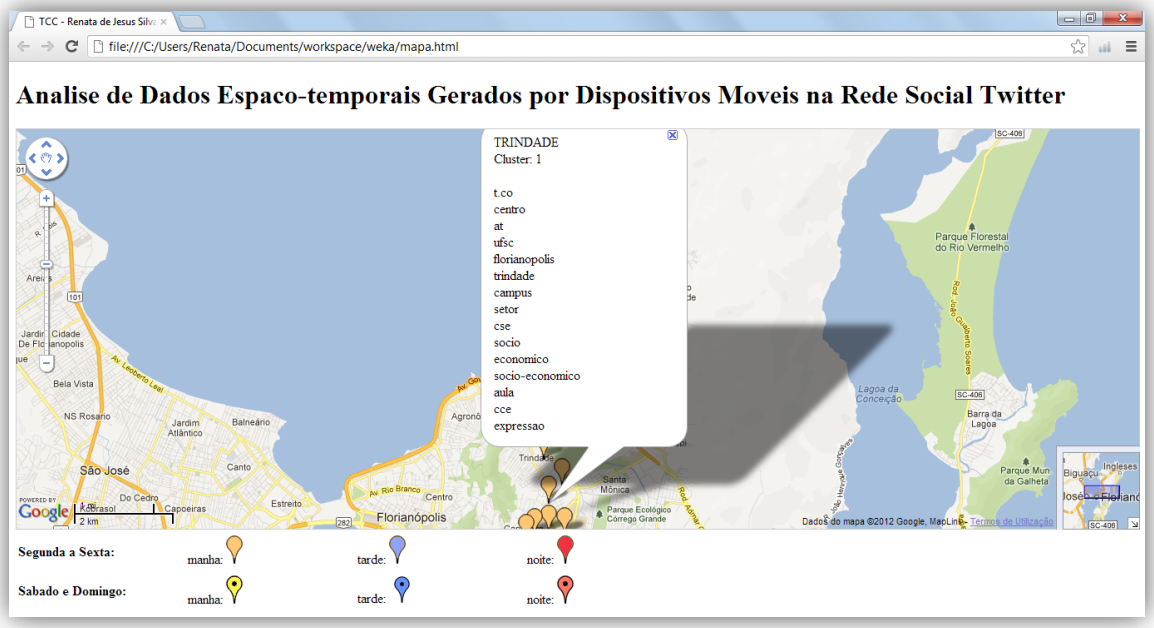


Figura 9 - 15 palavras mais frequentes do *cluster* 1 – bairro Trindade.

4.4 ESCOLHA DE ALGORITMO E TÉCNICA DE MINERAÇÃO DE DADOS

Esta seção discute sobre a técnica e o algoritmo utilizado neste projeto.

4.4.1 Preparação dos Dados Conforme Objetivos do Trabalho

A partir dos dados coletados no trabalho antecessor, foi criada uma tabela contendo uma amostragem referente a todos os registros de Florianópolis. Para popular esta tabela, somente foram utilizados dados originados por usuários que tuitaram, pelo menos, 10 vezes (Anexo A – DML 3).

Para facilitar o trabalho de mineração de dados com o conjunto de dados do Twitter, a partir do campo de data, que é do tipo *timestamp*, foi recuperado o dia da semana e o período (faixa de horário) do *tweet*. Assim, foram adicionadas duas colunas na tabela contendo estas informações.

Para a extração da informação, foram utilizadas funções disponíveis no PostgreSQL (Anexo A – DML 4 e 5). Além disso, foi determinado que em um dia existem quatro

períodos: madrugada, manhã, tarde e noite. Para isto foram definidas quatro faixas de horário, conforme a Tabela 3.

Faixa de horário	Período
0h ≥ **hora** < 6h	Madrugada
6h ≥ **hora** < 12h	Manhã
12h ≥ **hora** < 18h	Tarde
18h ≥ **hora** < 24h	Noite

Tabela 3 - Representação de períodos do dia conforme faixa de horário.

Exemplo da manipulação de datas: se a data é "2011-04-15 21:30:24" a coluna do mês recebe o valor "04" (abril), a coluna do dia da semana recebe o valor "sexta" e a coluna período recebe o valor "noite". Para esta pesquisa, decidiu-se juntar as faixas de horário "noite" e "madrugada" em um único período: noite.

Foi implementada a funcionalidade no Weka que permite visualizar as palavras mais frequentes em cada *cluster*. Para isto, foi necessário preparar o texto e utilizar um sistema de busca que existe no próprio SGBD utilizado – optou-se por utilizar esta ferramenta já que busca por cadeias de caracteres nas colunas de textos dos *tweets* tornaria a busca computacionalmente custosa. Com este sistema adotado, foi possível obter resultado de busca em texto mais veloz.

Para a preparação do texto do *tweet* em si, decidiu-se eliminar a acentuação ortográfica (Anexo A – DDL 4) para que a busca por texto encontrasse mais ocorrências de uma mesma palavra. Para tanto, "Florianópolis" e "Florianopolis" é a mesma palavra para o sistema.

O sistema de busca do PostgreSQL utilizado foi o Tsearch2[9]. Para utilizar este sistema é necessário optar por um dicionário. Este dicionário é responsável por definir a linguagem que o motor de busca irá trabalhar e, além disso, este pode executar a busca em texto baseando-se em um algoritmo, por exemplo, o algoritmo de *stemming* – este algoritmo visa encontrar variações de uma palavra. Para este estudo, o dicionário adotado foi o "simple" (Anexo A – DML 6), tal que neste dicionário o contador de palavras é feito pela palavra completa. Por exemplo, os termos "casa" e "casarão" serão considerados distintos pelo motor

9 Documentação disponível em: http://www.postgresql.org/docs/8.4/static/tsearch2.html . Último acesso em novembro 2012.

de busca (já o algoritmo de *stemming* trataria de outra maneira. Estas duas palavras seriam contabilizadas como a mesma, pois para ele as palavras são relacionadas).

No dicionário do Tsearch2, é possível adicionar palavras para ser ignoradas pelo motor de busca. O arquivo que contém estas palavras é chamado de *stopwords*. Para este estudo foram adicionadas ao dicionário padrão do Tsearch2 ("portuguese.stop") palavras indesejáveis para a busca (Anexo B). Além disso, o alfabeto também foi adicionado a esta lista.

Por fim, o Tsearch2 permite indexar os textos dos *tweets* tornando a busca mais rápida (Anexo A – DDL 5). O Tsearch2 utilizado não é case sensitive, isto é, não importa se a palavra digitada está em caixa alta ou baixa, por exemplo, "Florianópolis" e "florianópolis" é a mesma palavra para o sistema. A Tabela 4 apresenta uma amostra de como o Tsearch2 faz a indexação. A coluna "text" contém o texto do *tweet* e a coluna "vectors" contém este texto indexado pelo Tsearch2.

text	vectors
Que sono, que cansaço.	'cansaco':4 'sono':2
Tem deveres pra amanha?	'amanha':4 'deveres':2 'pra':3
Pq ela NAO responde a minha pergunta?	'nao':3 'pergunta':7 'pq':1 'responde':4
booom diaaaaaa !	'booom':1 'diaaaaaa':2
prof de filosofia é malucoo	'filosofia':3 'malucoo':5 'prof':1
@luholiveira Parabéns pelo dia dos Designers!	designers':6 'dia':4 'luholiveira':1 'parabens':2
Minha mochila ta muito pesada, socorro	'mochila':2 'pesada':5 'socorro':6 'ta':3

Tabela 4 – Comparação do texto antes e depois de ser indexado pelo Tsearch2.

A busca em texto foi quantitativa e não qualitativa, ou seja, o importante foi o número de vezes que a palavra foi utilizada e não o seu significado ou se o texto estava gramaticalmente correto (um exemplo de busca em texto pode ser visto no Anexo A – DML 7). Ressaltando, a única consideração empregada foi a eliminação de palavras contidas no arquivo de *stopwords*, pois estas foram consideras irrelevantes e, até mesmo, prejudiciais em uma análise futura.

4.5 LIMPEZA DE DADOS

Somente dados com origem na cidade de Florianópolis foram utilizados para esta pesquisa. Além disso, desconsiderou-se da base de estudo dados de usuários que postaram no Twitter menos de 10 vezes.

Entre 35 bairros de Florianópolis, 15 foram desconsiderados para a pesquisa. Os ignorados possuem número pequeno de registros na base de dados. Uma relação da quantidade de *tweets* por bairro e a percentagem que esta representa (referente aos 20 bairros analisados neste estudo) podem ser vistas no gráfico da Figura 10.

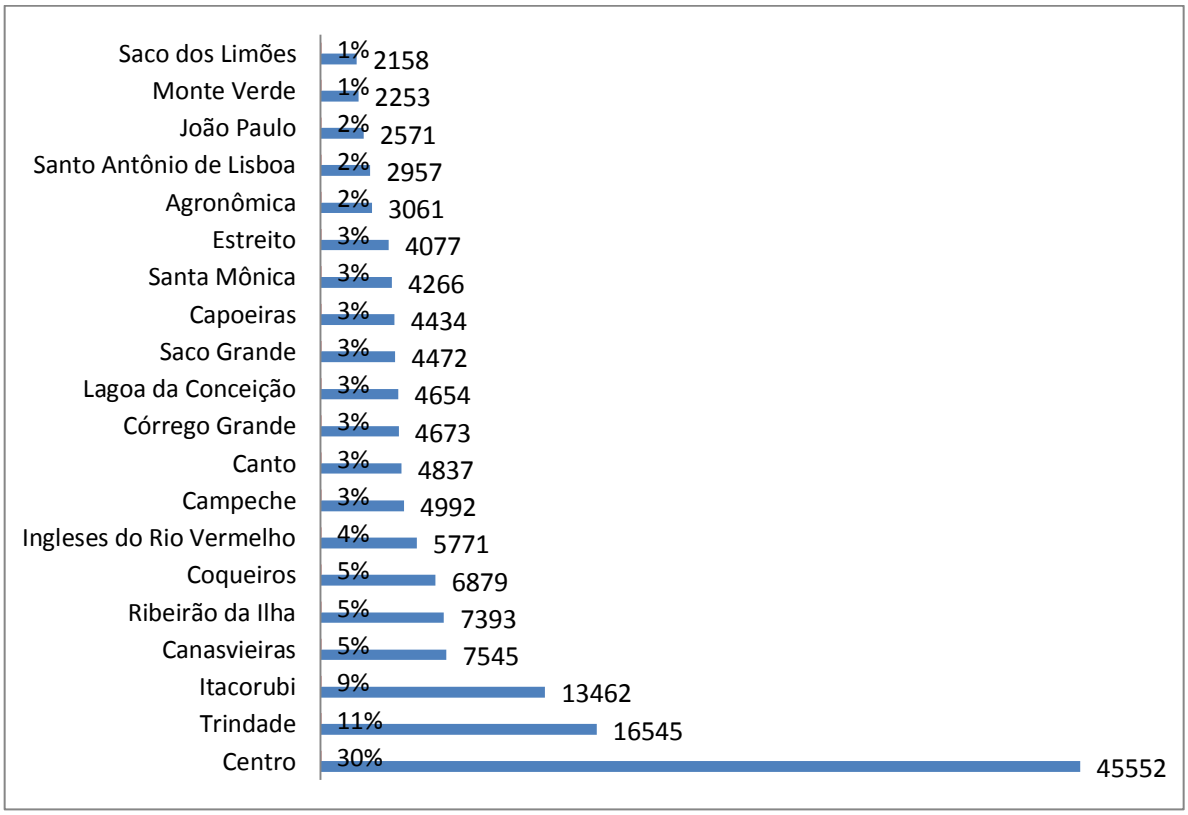


Figura 10 - Bairros com a quantidade de tweets e a percentagem que esta representa.

4.6 SELEÇÃO DOS DADOS

Como o algoritmo DBSCAN da ferramenta Weka não conhece dados de tipo geográficos, a solução encontrada foi trabalhar com os campos “latitude” e “longitude” da base. Portanto, os atributos selecionados da tabela de *tweets* de Florianópolis foram estes. Assim, o algoritmo é capaz de agrupar coordenadas geográficas próximas umas as outras com precisão, sem a necessidade de transformar os dados antes de começar o processo de mineração de dados.

O conjunto de dados foi filtrado por 3 atributos: bairro, período do dia (manhã ou tarde ou noite) e dia da semana (segunda a sexta-feira ou sábado e domingo), totalizando 6 consultas por bairro. Assim, utilizando o bairro Trindade como exemplo, as consultas foram:

Consulta 1:

Bairro: Trindade	**Período:** Manhã	**Dia da semana:** Segunda a Sexta-feira

```
SELECT
	lat, long
FROM
	sc_floripa_mais10_tweetsbrasil
WHERE
	bairro = 'TRINDADE'
	AND (dia_semana = 'segunda' or dia_semana = 'terça' or dia_semana = 'quarta' or dia_semana = 'quinta' or dia_semana = 'sexta')
	AND periodo = 'manhã'
ORDER BY
	datetime;
```

Consulta 2:

Bairro: Trindade	**Período:** Tarde	**Dia da semana:** Segunda a Sexta-feira

```
SELECT
	lat, long
FROM
	sc_floripa_mais10_tweetsbrasil
WHERE
	bairro = 'TRINDADE'
	AND (dia_semana = 'segunda' or dia_semana = 'terça' or dia_semana = 'quarta' or dia_semana = 'quinta' or dia_semana = 'sexta')
	AND periodo = 'tarde'
ORDER BY
	datetime;
```

Consulta 3:

Bairro: Trindade	**Período:** Noite	**Dia da semana:** Segunda a Sexta-feira

```
SELECT
        lat, long
FROM
        sc_floripa_mais10_tweetsbrasil
WHERE
        bairro = 'TRINDADE'
        AND (dia_semana = 'segunda' or dia_semana = 'terça' or dia_semana = 'quarta' or
dia_semana = 'quinta' or dia_semana = 'sexta')
        AND (periodo = 'noite' or periodo = 'madrugada')
ORDER BY
        datetime;
```

Consulta 4:

Bairro: Trindade	**Período:** Manhã	**Dia da semana:** Sábado e Domingo

```
SELECT
        lat, long
FROM
        sc_floripa_mais10_tweetsbrasil
WHERE
        bairro = 'TRINDADE'
        AND (dia_semana = 'sábado' or dia_semana = 'domingo')
        AND periodo = 'manhã'
ORDER BY
        datetime;
```

Consulta 5:

Bairro: Trindade	**Período:** Tarde	**Dia da semana:** Sábado e Domingo

```
SELECT
        lat, long
FROM
        sc_floripa_mais10_tweetsbrasil
WHERE
        bairro = 'TRINDADE'
        AND (dia_semana = 'sábado' or dia_semana = 'domingo')
        AND periodo = 'tarde'
ORDER BY
        datetime;
```

Consulta 6:

Bairro: Trindade	**Período:** Noite	**Dia da semana:** Sábado e Domingo

```
SELECT
        lat, long
FROM
        sc_floripa_mais10_tweetsbrasil
WHERE
        bairro = 'TRINDADE'
        AND (dia_semana = 'sábado' or dia_semana = 'domingo')
        AND (periodo = 'noite' or periodo = 'madrugada')
ORDER BY
        datetime;
```

4.7 MINERAÇÃO DE DADOS: AGRUPAMENTO (*CLUSTER*)

Para executar o algoritmo DBSCAN no Weka é necessário indicar 3 valores para os atributos épsilon, mínimo de pontos e tipo de distância. A escolha destes valores são mostrados na Tabela 5.

Atributo	Valor
Epsilon	0,045.
MinPoints	2,5% da quantidade de dados da consulta. O limite mínimo é 12.
DistanceType	Euclidiana.

Tabela 5 - Parâmetros utilizados no DBSCAN.

A Figura 11 apresenta parte da saída gerada pela execução do algoritmo DBSCAN no Weka.

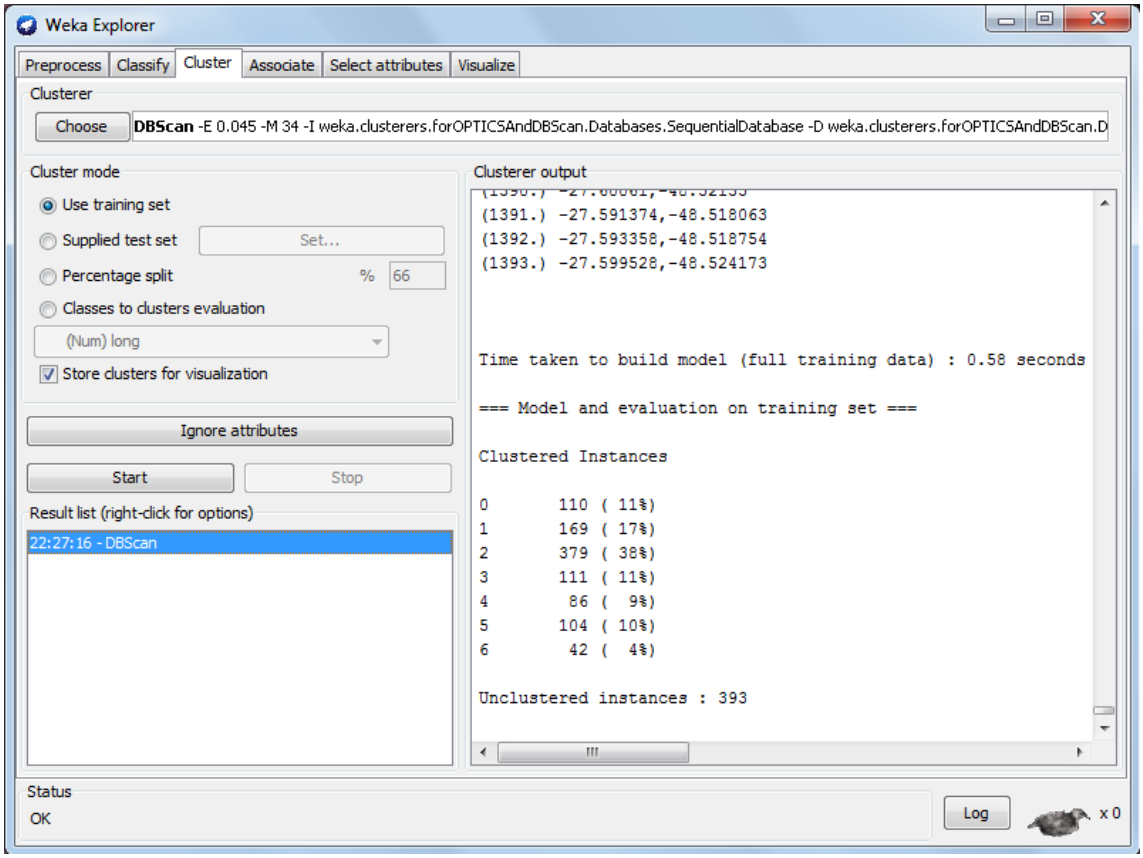


Figura 11 - Parte da saída gerada pelo algoritmo DBSCAN.

Após executar as consultas referentes aos bairros de Florianópolis, os centróides podem ser visualizados na Figura 12.

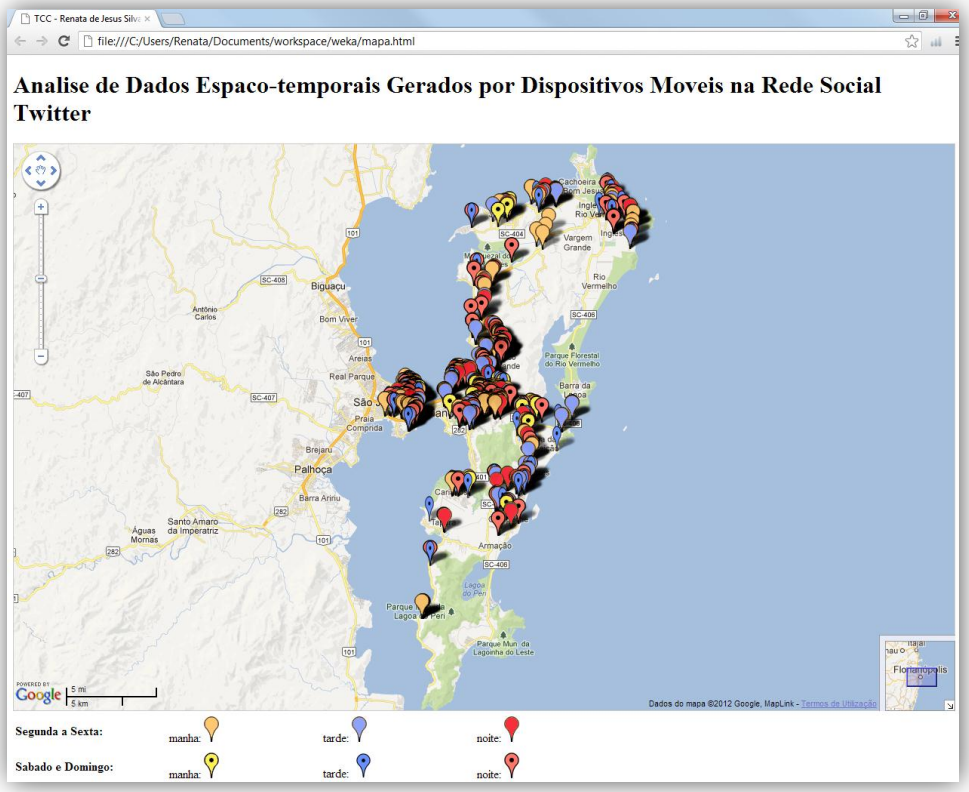


Figura 12 - Visualização do mapa completo.

4.8 AVALIAÇÃO DOS RESULTADOS ENCONTRADOS

De acordo com as consultas geradas e parâmetros utilizados no DBSCAN, foi possível analisar os bairros conforme as características espaço-temporais. A localização foi obtida através das coordenadas geográficas latitude e longitude e a medida temporal foi definida por dia da semana e turno deste dia.

As análises seguintes foram realizadas a partir das 6 consultas, por bairro, aplicadas ao *software* Weka. Esta seção apresenta uma análise geral dos bairros (organizados em ordem decrescente da quantidade de *tweets* por bairro) selecionados para a pesquisa. Para cada bairro, existem quatro informações relevantes:

- Area: disponível no site da Prefeitura Municipal de Florianópolis, na unidade de medida km²;
- *Tweets*: quantidade de *tweets* registrada no bairro;
- Usuários distintos: quantidade de usuários distintos que mandaram *tweets* no bairro;
- Descrição da análise: breve descrição considerando o comportamento dos dados após a mineração de dados.

1) Centro

Área: 5.368 km²	**Tweets**: 45552	**Usuários Distintos:** 1108

Descrição da análise: A maioria das consultas gerou somente um *cluster* situado no meio deste bairro, mais especificamente, no meio da Avenida Rio Branco. Este resultado foi encontrado devido à imensa quantidade de registros espalhados no Centro (Figura 13). Portanto, os centróides ficaram exatamente no meio do bairro porque os dados estavam geograficamente homogêneos. Para que o algoritmo DBSCAN pudesse gerar mais *clusters*, o ideal, neste caso, é diminuir o valor do épsilon para encontrar grupos formados em determinados locais populares como bares, restaurantes, centros comerciais, praças, supermercados, etc. Além deste comportamento, é importante ressaltar que o Centro é o bairro que mais possui *tweets* (30% do total da base de Florianópolis) e, portanto, houve consultas que possuíam mais de 10 mil dados. A Figura 13 possui os *tweets* plotados no mapa, visualizado pela ferramenta Quantum GIS. Estes dados são somente do bairro Centro no período da tarde no intervalo de segunda a sexta-feira, totalizando 11.315 registros.

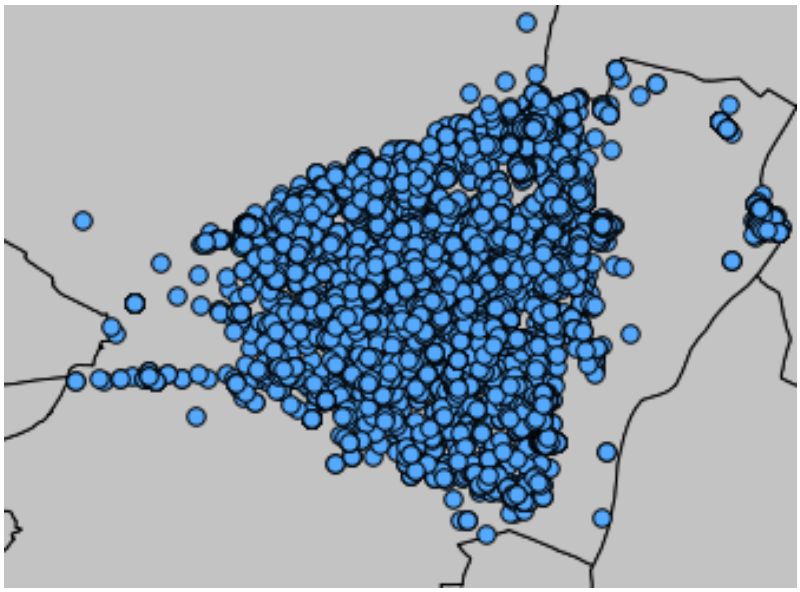


Figura 13 - Mapa do bairro Centro com *tweets* visualizado pela ferramenta Quantum GIS.

Na Figura 14 pode ser visualizado o bairro Centro com os *clusters* formados, ressaltando que o marcador de cor azul representa o centróide do *cluster* formado com os mesmos parâmetros utilizados no exemplo da Figura 13.

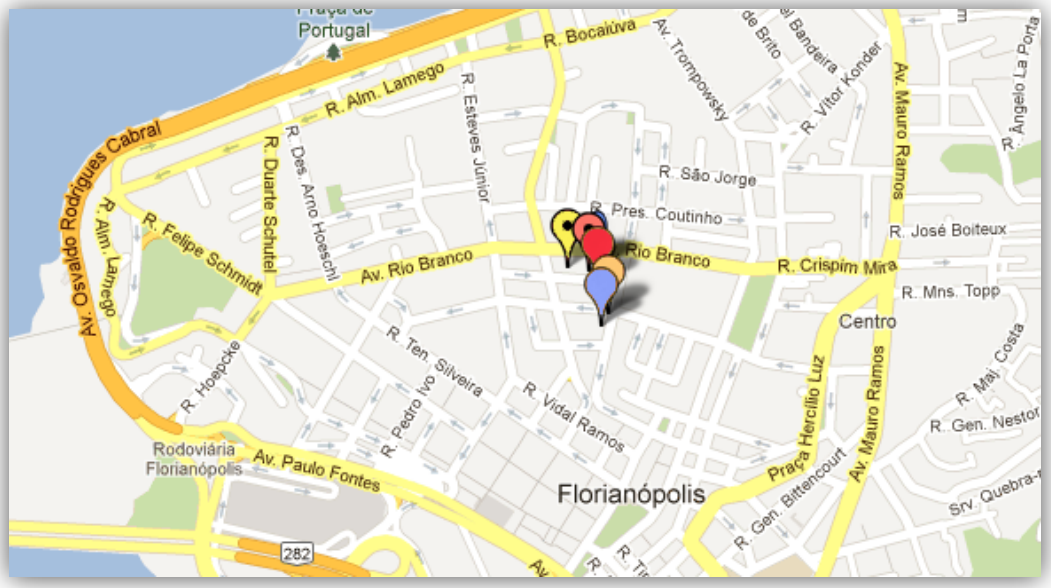


Figura 14 - *Clusters* formados no bairro Centro.

1.1) Centro – outras análises

Como o bairro Centro obteve um número grande de registro de *tweets*, optou-se por explorar mais esta região a fim de detectar locais conhecidos enriquecendo a pesquisa.

O algoritmo DBSCAN foi executado com diferentes valores de atributos, épsilon e mínimo de pontos, até se tornar possível a descoberta de locais de interesse. Dois destes experimentos são detalhados nas seções seguintes.

1.1.1) Resultados encontrados – *experimento 1*

Mínimo de pontos = 1,25% e Épsilon = 0,011.

Em relação às análises dos demais bairros, o número mínimo de pontos utilizado foi reduzido pela metade e o épsilon representou 25% do valor. Com estes parâmetros, foi possível identificar locais de interesse como: (i) Terminal de Integração do Centro (TICEN) – períodos manhã e tarde de dia de semana e noite tanto de dia de semana quanto de fim de semana; (ii) Intituto Estadual de Educação (IEE) – manhã e tarde de dia de semana; (iii) Praça XV de Novembro – manhã de fim de semana; (iv) Catedral Metropolitana de Florianópolis – tarde e noite de fim de semana; (v) Beiramar Shopping – manhã e tarde de dia de semana e fim de semana; (vi) Boate El Divino – tarde e noite de fim de semana; (vii) Boate 1007 – manhã e noite de fim de semana; (viii) Mercado Público – tarde de fim de semana; (ix) Morro

da Cruz – manhã de fim de semana; (x) Instituto Federal de Santa Catarina (IF-SC) – tarde de dia de semana; (xi) Centro executivo localizado na Avenida Mauro Ramos – manhã de dia de semana, etc.

Com o mapa ilustrado na Figura 15 (gerado pela execução do *experimento 1*) é possível identificar alguns locais citados.

Figura 15 - Parte do resultado do *Experimento 1*.

1.1.2) Resultados encontrados – *experimento 2*

Mínimo de pontos = 45 e Épsilon = 0,0005.

Para este experimento, foi reduzido radicalmente o épsilon com objetivo de detectar locais de áreas menores, já que o raio definido foi bem pequeno. Assim, foi possível evitar que pontos em diferentes locais permanecessem em um mesmo *cluster*. O número mínimo de pontos ficou fixo neste experimento. Com isto, pode-se analisar o comportamento de usuários que tuitaram em um local específico.

Em relação à análise anterior – *experimento 1* – foi observado que: (i) na Praça XV de Novembro (que foi considerada uma região densa no experimento anterior),

por possuir uma área relativamente grande, não foi identificada como uma região densa, justamente por os *tweets* estarem mais distantes entre si neste local; (ii) alguns locais não detectados com análises anteriores puderam ser encontrados nesta análise, como a Universidade do Sul de Santa Catarina (UNISUL) e Terminal Rodoviário Rita Maria. Alguns *clusters* em locais residenciais podem ser obervados, por exemplo, nas ruas Arno Hoeschl e Esteves Junior.

Com o mapa ilustrado na Figura 16 (gerado pela execução do *experimento 2*) é possível identificar alguns locais citados, podendo ser observado *clusters* em diferentes pontos, que são: Terminal Rita Maria e UNISUL.

Figura 16 – Parte do resultado do *experimento 2*.

2) Trindade

Área: 3.502 km²	**Tweets**: 16545	**Usuários Distintos:** 530

Descrição da análise: Todas as consultas pertencentes a este bairro possuem *cluster* situado no Terminal de Integração da Trindade (TITRI).

O que pode ser bem observado é que, de segunda a sexta-feira, nos períodos manhã e tarde, houve ocorrências de *clusters* dentro da UFSC, onde o mais denso é no turno da tarde e próximo à Biblioteca Universitária. Ainda, durante a semana e no período da noite, surgiu mais um *cluster* dentro da Universidade, mais especificamente no Centro Tecnológico (CTC). Grupo formado entre segunda e sexta-feira tem textos relacionados à aula. Em final de semana não há ocorrência de *cluster* na UFSC, porém formou-se um *cluster* próximo ao bar CSC (próximo à UFSC).

3) Itacorubi

Área: 12.756 km²	**Tweets**: 13462	**Usuários Distintos:** 427

Descrição da análise: O comportamento comparando tanto no período do dia como em dia de semana, não ficou muito distinto uns dos outros. Os centróides dos *clusters* praticamente ficaram no mesmo local.

4) Canasvieiras

Área: 29.125 km²	**Tweets**: 7545	**Usuários Distintos:** 344

Descrição da análise: *Clusters* geograficamente bem distribuídos. *Clusters* na praia de Jurerê possuem muitos *tweets* relacionados à praia, *open shopping* e restaurantes.

5) Ribeirão da Ilha

Área: 52.565 km²	**Tweets**: 7393	**Usuários Distintos:** 544

Descrição da análise: Em todas as consultas, os *clusters* próximos ao Aeroporto Internacional Hercílio Luz ficaram com mais de 70% dos dados. Na tarde de fim de semana, houve um *cluster* no estádio Aderbal Ramos da Silva ("Ressacada") do time de futebol Avaí. Neste bairro, no geral, os *clusters* ficaram distantes uns dos outros.

6) Coqueiros

Área: 1.751 km²	**Tweets**: 6879	**Usuários Distintos:** 246

Descrição da análise: No geral, consulta gerou *clusters* com quantidade de dados bem distribuídos. No geral, os grupos formados estão em pontos comuns independente do período do dia e dia da semana. Isto pode ser observado na Figura 17.

Figura 17 - *Clusters* formados no bairro Coqueiros.

7) Ingleses do Rio Vermelho

Área: 19.456 km²	**Tweets**: 5771	**Usuários Distintos:** 210

Descrição da análise: Não houve muitas diferenças no comportamento dos grupos formados. Houve casos em que só formaram áreas densas entre segunda e sexta-feira e, nestes casos, analisando os textos, as palavras são referentes à aula.

8) Campeche

Área: 34.863 km²	**Tweets**: 4992	**Usuários Distintos:** 198

Descrição da análise: Não houve diferença no comportamento dos grupos. Exceto consultas que envolvem registros do período manhã, este bairro formou vários *clusters* (cerca de 10 *clusters* por consulta). Analisando as palavras mais frequentes, podem-se perceber grupos formados na Avenida Pequeno Príncipe, pizzaria e posto de gasolina.

9) Canto

Área: 0.680 km²	**Tweets**: 4837	**Usuários Distintos:** 207

Descrição da análise: Os *clusters* formados ficaram bem próximos uns dos outros. Isto está ligado à geografia deste local (este bairro possui menos de 1 km² de área). No estádio Orlando Scarpelli, do time de futebol Figueirense, foram formados c*lusters* no período da noite tanto em dia de semana quanto em fim de semana.

10) Córrego Grande

Área: 6.603 km²	**Tweets**: 4673	**Usuários Distintos:** 287

Descrição da análise: Os grupos tiveram comportamentos parecidos independente do período do dia ou dia da semana, exceto os *clusters* formados no centro de Engenharia Química de Alimentos, onde houve ocorrência de *clusters* somente de segunda a sexta-feira.

11) Lagoa da Conceição

Área: 53.833 km²	**Tweets**: 4654	**Usuários Distintos:** 425

Descrição da análise: De segunda a sexta-feira, os maiores *clusters* formados são no Terminal de Integração da Lagoa da Conceição (TILAG), o restante de *clusters*, no geral, tem poucos *tweets*. Também foram formados grupos no mirante da lagoa, bares e restaurantes.

12) Saco Grande

Área: 11.016 km²	**Tweets**: 4472	**Usuários Distintos:** 263

Descrição da análise: No geral, tarde e noite de dia e fim de semana tem o mesmo comportamento. Formaram *clusters* próximos e bem distribuídos quanto à quantidade de registros em cada grupo.

13) Capoeiras

Área: 2.816 km²	**Tweets**: 4434	**Usuários Distintos:** 314

Descrição da análise: A Figura 18 permite visualizar que o bairro de Capoeiras formou *clusters* com centróides comuns. O *cluster* marcado com o círculo azul está localizado no supermercado Angeloni, o verde situa-se em uma empresa de Tecnologia da Informação – Dígitro. A marcação vermelha está localizada próxima a uma delegacia e a uma pizzaria. Tanto o *cluster* destacado pelo círculo amarelo como de cor laranja não são locais comerciais, consultando no banco de dados, pode-se perceber que nestes dois casos os registros dos grupos estão associados a poucos usuários (menos de 5 usuários distintos).

Figura 18 - *Clusters* formados no bairro Capoeiras.

14) Santa Mônica

Área: 0.590 km²	**Tweets**: 4266	**Usuários Distintos:** 454

Descrição da análise: No geral, grupos que foram formados no *shopping* Iguatemi representam entre 30 e 40% dos registros. Podem ser observado *clusters* ao longo da Avenida Madre Benvenuta.

15) Estreito

Área: 1.388 km²	**Tweets**: 4077	**Usuários Distintos:** 212

Descrição da análise: Formaram-se muitos *clusters* em todas as consultas, exceto manhã de fim de semana. Os *Clusters* ficaram próximos uns dos outros.

16) Agronômica

Área: 1.964 km²	**Tweets**: 3061	**Usuários Distintos:** 393

Descrição da análise: Formaram-se muitos *clusters* em todas as consultas, exceto nas manhãs de final de semana que não foi formado nenhum. Geograficamente, *clusters* bem distribuídos entre si. Quantidade de dados por *cluster* bem distribuído também.

17) Santo Antônio de Lisboa

Área: 21.527 km²	**Tweets**: 2957	**Usuários Distintos:** 313

Descrição da análise: No geral, as consultas formaram muitos clusters. Durante a semana, *clusters* com *tweets* relacionados ao centro empresarial Corporate Park e *tweets* relacionados à aula. Final de semana e dia semana no período da noite: *cluster* formado na casa noturna Stage Music Park.

18) João Paulo

Área: 2.804 km²	**Tweets**: 2571	**Usuários Distintos:** 142

Descrição da análise: No geral, as consultas formaram muitos *clusters*. *Clusters* distantes entre si. Com relação a período ou dia semana, o comportamento dos usuários mostrou-se parecido.

19) Monte Verde

Área: 5.054 km²	**Tweets**: 2253	**Usuários Distintos:** 247

Descrição da análise: Maioria dos *clusters* formados ficou próximo ao Floripa Shopping.

20) Saco dos Limões

Área: 3.895 km²	**Tweets**: 2158	**Usuários Distintos:** 130

Descrição da análise: Ocorreram muitos locais com *clusters* formados apenas em dia de semana.

4.9 EXTRAÇÃO DE CONHECIMENTO

Como o algoritmo utilizado neste projeto foi o DBSCAN e o mesmo tem por característica indicar grupos em regiões densas, há uma grande possibilidade de os pontos centrais gerados pelos *clusters* estejam sobre, ou muito próximos a, locais atrativos. Por exemplo, universidades, restaurantes, bares, *shopping centers*, estádio de futebol, centros comerciais, empresas, entre outros. Isto pode ser percebido ao visualizar os marcadores plotados pelo Weka no mapa do Google Maps desenvolvido neste trabalho.

Com as figuras seguintes é possível observar exemplos de formação de *clusters*. Em cada figura estão destacados (circulados em vermelho ou verde) os *clusters* em regiões que possuem locais atrativos.

Com a Figura 19, é possível identificar que no bairro Santa Mônica foram formados *clusters* em locais populares. A primeira região bastante densa (tardes e noites) foi formada no *shopping center* Iguatemi (na figura, este *shopping* está circulado em vermelho). A segunda região densa está localizada em uma região que possui um centro comercial. E nesta, formaram-se *clusters* somente em dias úteis.

Figura 19 - *Clusters* formados no bairro Santa Mônica.

Na Figura 20 é possível verificar que há grande quantidade de *clusters*, formados em dias úteis, dentro da UFSC.

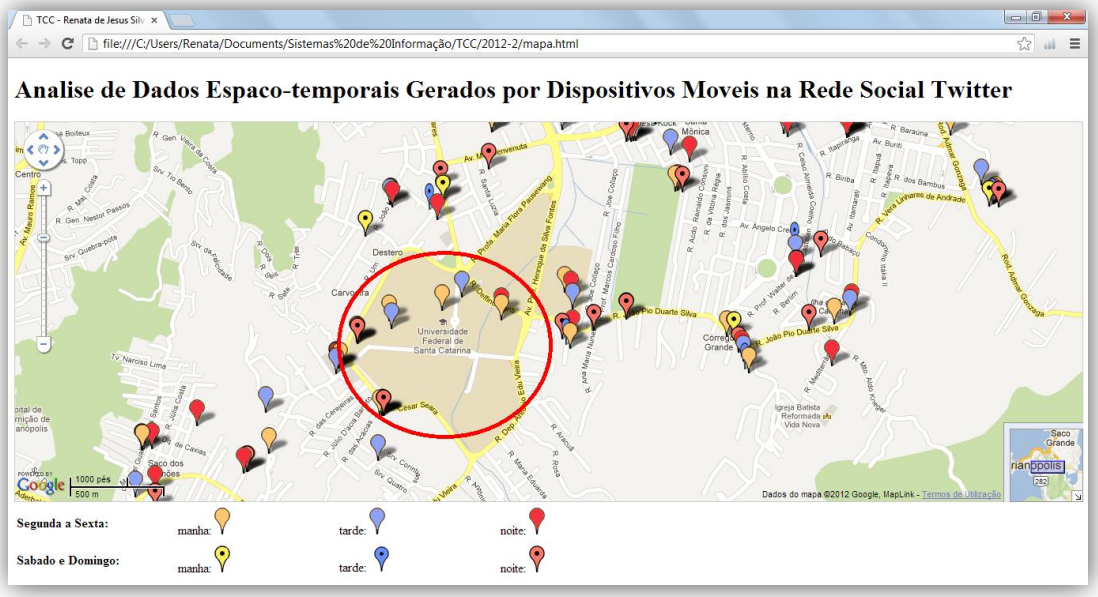


Figura 20 - *Clusters* formados na UFSC – bairro Trindade.

Na Figura 21 podem ser visualizados os grupos formados no Terminal de Integração da Lagoa (TILAG), no bairro Lagoa da Conceição (destacados pelo círculo vermelho). Observa-se que neste local não há formação de *cluster* à noite, apenas pela manhã e tarde, tanto nos dias de semana quanto nos finais de semana.

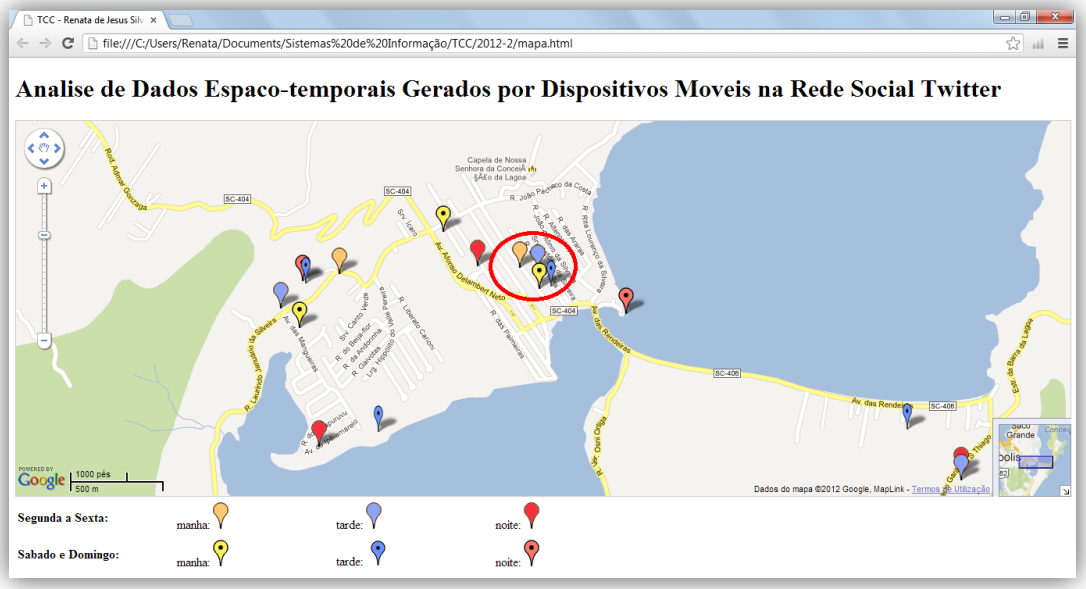


Figura 21 - *Clusters* formados no bairro Lagoa da Conceição, em destaque o TILAG.

Os grupos formados pelos *clusters* da Figura 22 (circulado em vermelho) estão situados no estádio de futebol Orlando Scarpelli. Estes *clusters* correspondem principalmente a *tweets* postados nas tardes e noites de fim de semana, o que coincide com a maioria dos jogos que são sábados a noite e domingos a tarde.

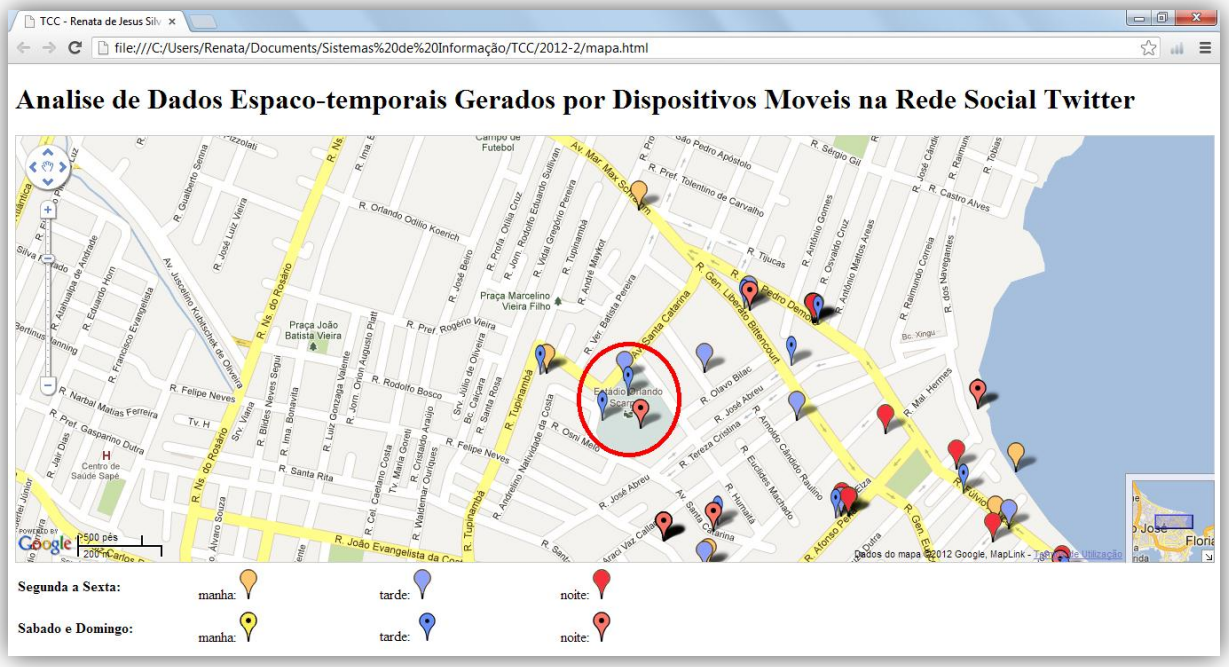


Figura 22 - *Clusters* formados no estádio Orlando Scarpelli – bairro Canto.

Em suma, o comportamento do homem na sua rotina diária pode ser observado diretamente com a análise dos usuários do Twitter. Por exemplo, comumente, locais que determinam escolas e faculdades são pontos de interesse durante os dias úteis da semana e, nos finais de semana não ocorrem *clusters* nesses locais.

As 15 palavras mais utilizadas, na maioria dos *clusters*, indicam o bairro em que o *tweet* situa-se, o nome da rua e palavras com correspondência direta ao nome do local. A palavra, que referencia um *site*, "4sq.com"[10] também ocorre em grande parte dos casos. Uma maneira de entender melhor o que os usuários mais comentam por *cluster*, poderia ser buscando mais palavras frequentes.

10 4sq.com (Foursquare) é um *microblog* que permite indicar onde o usuário está localizado. Último acesso em novembro 2012.

Nas manhãs de fim de semana e noites de dia de semana, nas palavras frequentes, é bem evidente aparições de palavras que sugerem que o usuário esteja em casa. Por exemplo, foram encontradas tais palavras nestes tipos de *clusters*: dormir, casa, mãe, sono.

Analisando de um modo geral, pode-se perceber que, na maioria dos casos, consultas com parâmetros de período "manhã" possuem um menor número de registros, inclusive, algumas consultas deste período combinados com fim de semana não formaram *clusters*.

O comportamento dos *clusters* gerados no bairro Centro foi diferente do comportamento do restante dos bairros analisados, pois estas consultas geraram, na maioria dos casos, somente um *cluster* situado no meio deste bairro. Isto aconteceu pela imensa quantidade de *tweets* registradas no bairro.

Há bairros que os usuários se comportam de maneiras bem diferentes se comparado dia de semana e fim de semana. Isto, na maior parte dos casos, deve-se ao fato de haver muitos *clusters* formados em colégios e faculdades. A Trindade é um bairro que contem este exemplo, pois foram formados vários *clusters* dentro da UFSC, em se tratando de registros durante a semana. Nos finais de semana, a UFSC não registrou nenhum *cluster*.

5. CONSIDERAÇÕES FINAIS

5.1 CONCLUSÃO

Mineração de dados espaço-temporais, com foco em detecção de agrupamentos, em redes sociais não é um assunto muito explorado. Esta pesquisa buscou conhecer o comportamento dos usuários do Twitter – especificamente na cidade de Florianópolis. De acordo com o conhecimento extraído, podem-se tirar conclusões do interesse da população e, analisar quais as opiniões delas sobre determinadas localizações. Por exemplo, donos de empresas, tendo acesso a estas informações, podem analisar a satisfação de colaboradores e/ou clientes. Este tipo de pesquisa poderá contribuir, por exemplo, com pesquisas de *marketing*, consequentemente, há maiores chances de obter resultados seguros.

Este trabalho de conclusão de curso trouxe um enorme conhecimento nas áreas de banco de dados geográficos, manipulação com dados espaço-temporais e mineração de dados. Os dois primeiros não foram assuntos abordados durante o curso. No entanto, não houve dificuldades em entender os conceitos e como deveria ser aplicado na prática, pois – de forma indireta – estes assuntos tiveram como base assuntos estudados, como “Banco de Dados”, “Mineração de Dados” e “Sistemas Inteligentes”. Destaco esta última disciplina como base para desenvolvimento de algoritmos de mineração de dados.

Para atender a proposta do projeto, foi adaptado o código da ferramenta Weka. Isto tornou o *software* uma excelente ferramenta também para visualização. Desta maneira, o resultado encontrado pelo algoritmo DBSCAN pode ser melhor analisado.

O resultado final das análises evidenciou que o comportamento dos usuários do *microblog* é equivalente ao comportamento do homem em sua rotina diária. Foi possível identificar bairros que possuem mais locais atrativos e bairros que se limitam a somente uma região densa. Por exemplo, o Ribeirão da Ilha, que é o 5° bairro de Florianópolis de maior área, possui apenas uma região atrativa, que é o aeroporto de Florianópolis. O restante dos grupos encontrados neste bairro foi, na maior parte, locais residenciais e não tiveram um número tão alto de *tweets* nestes como ocorreu no aeroporto.

5.2 TRABALHOS FUTUROS

Um *cluster* pode ser formado apenas por *tweets* de um usuário. Pode ser interessante, tratar os dados para evitar que sejam considerados *Stupid, Pointless Annoying Message* (SPAM), ou,

até mesmo, impedir a formação de *cluster* se nele não atingir um número mínimo de usuários. Além disso, outras sugestões são:

- Generalizar adaptações que foram feitas no código Weka para que este funcione em qualquer base de dados;
- Permitir ações de usuário no mapa, como: limpar *clusters* já populados no mapa; aplicar filtros para visualizar somente *clusters* de interesse. Em outros contextos, permitir também mais flexibilidade ao usuário, adicionandos componentes na interface gráfica para isto;
- Detectar se em algum dia, em um determinado local e horário, ocorreu algum comportamento fora do padrão;
- Aplicar o algoritmo DBSCAN em diferentes níveis de abstração, isto é, variar os valores de épsilon e mínimo de pontos em cada bairro em particular. Frizando que, para este estudo, foram utilizados parâmetros fixos para todos os bairros, independente de área ou quantidade de *tweets* que estes possuem (com exceção do bairro Centro). Desta maneira, a mineração de dados poderá ser mais aprofundada permitindo a descoberta de informação potencialmente útil.

Este trabalho apresentou uma análise bem geral e preliminar dos *tweets* de Florianópolis. Extrair conhecimento mais específico e aprofundado pode ser possível quando houver um objetivo bem definido para a mineração.

REFERÊNCIAS

BAJPAI, Aman; LITORIYA, Ratnesh; SHARMA, Narendra**. Comparison the various clustering algorithms of weka tools**. International Journal of Emerging Technology and Advanced Engineering. ISSN 2250-2459, Volume 2, Issue 5, May 2012.

BHAT, F. et al. **A Software System for Data Mining with Twitter**. 10th IEEE International Conference on Date of Conference. Londres, set 2011.

COSTA, Nabucodonosor Coutinho**. Indexação de textos com o Tsearch2 – Busca Bonita e Veloz**. Revista Linux Magazine Online disponível em www.linuxnewmedia.com.br/images/uploads/pdf_aberto/LM23_postgresql.pdf. Visualizado em outubro, 2012.

EGENHOFER, Max; HERRING, John. **Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases**. 1991.

FILKOV, V.; KIENA S. **Integrating microarray data by consensus clustering**. International Journal on Artificial Intelligence Tools, 13(4): 863–880, 2004.

GUO, Diansheng; MENNIS, Jeremy. Spatial data mining and geographic knowledge discovery-An introduction. **Computers, Environment and Urban Systems**. 2009. p. 403-408.

HAN, Jiawei; KAMBER, Micheline. **Data Mining: Concepts and Techniques**. 2. ed. São Francisco, CA, 2006. 772 p.

IBGE. Instituto Brasileiro de Geografia e Estatística. **Documento compactado.** Disponível em: <http://www.ibge.gov.br/servicodados/Download/Download.ashx?u=geoftp.ibge.gov.br/mapas_estatisticos/censo_2010/mapas_de_setores_censitarios/SC/4205407.zip>. Último acesso em novembro 2012.

KEIM, Daniel; KISILEVICH Slava; MANSMANN, Florian. **P-DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos**. University of Konstanz.

LINK NACIONAL. **Script de Múltiplos Pontos.** Disponível em: <http://www.linknacional.com.br/criar-site/2011/01/google-maps-api-multiplos-pontos-no-mapa-openinfowindowhtml>. Último acesso em novembro 2012.

AILON, N. et al. **Aggregating incosistent information: ranking and clustering**. In Proceedings of the thirty-seventh anual ACM Symposium on Theory of Computing, 2005. p. 684-693.

XI, Bowei; XIA, Yuni. **Conceptual Clustering Categorical Data with Uncertainty Indiana University**. Purdue University Indianapolis Indianapolis, IN 46202, USA.

ANEXOS

ANEXO A - LISTA DE *QUERYS*

DML 1

Criação da coluna geométrica e conversão dos atributos latitude e longitude:

```
SELECT
        AddGeometryColumn('public', 'sc_floripa_mais10_tweetsbrasil', 'the_geom', 4326,
'POINT', 2)

UPDATE
        sc_floripa_mais10_tweetsbrasil
        SET the_geom = ST_SetSRID(ST_MakePoint(long, lat),4326)
```

DDL 1

Para unir os dados de maneira que se formem bairros, a *query* abaixo foi utilizada:

```
CREATE TABLE mapa_bairros_florianopolis AS
    SELECT
        cd_geocods,
        cd_geocodd,
        nm_distrit,
        cd_geocodm,
        nm_municip,
        nm_micro,
        nm_meso,
        ST_Union(the_geom) AS the_geom
    FROM
        uniaobairros
    GROUP BY
        cd_geocods, cd_geocodd,nm_distrit,cd_geocodm,nm_municip,nm_micro,nm_meso;
```

DDL 2

Criação das tabelas de usuário e *tweets* do Brasil:

```
CREATE TABLE sc_usersbrasil (
    id bigint NOT NULL,
    screen_name character varying(32) DEFAULT NULL::character varying,
    "name" character varying(64) DEFAULT NULL::character varying,
    fullprofile text,
    datetime timestamp without time zone DEFAULT now(),
    distancia_media double precision,
    distancia double precision,
    CONSTRAINT sc_usersbrasil_pkey PRIMARY KEY (id)
);

CREATE TABLE tweetsbrasil (
    id bigint NOT NULL,
    "text" character varying(150) NOT NULL,
    lat double precision NOT NULL,
    "long" double precision NOT NULL,
    datetime timestamp without time zone NOT NULL,
    created_at timestamp without time zone NOT NULL,
    user_id bigint NOT NULL,
    country character varying(128) DEFAULT NULL::character varying,
    full_place character varying(128) DEFAULT NULL::character varying,
    place_type character varying(8) DEFAULT NULL::character varying,
    state character varying(64) DEFAULT NULL::character varying,
    city character varying(64) DEFAULT NULL::character varying,
    the_geom geometry,
    CONSTRAINT sc_tweetsbrasil_pkey PRIMARY KEY (id),
    CONSTRAINT enforce_dims_the_geom CHECK (st_ndims(the_geom) = 2),
    CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) =
            POINT'::text OR the_geom IS NULL),
    CONSTRAINT enforce_srid_the_geom CHECK (st_srid(the_geom) = 4326)
);
```

DML 2

Operação de atualização utilizada para o cruzamento dos dados a fim de detectar o bairro de origem do *tweet:*

```
UPDATE
    tweets f set bairro = ( SELECT b.nm_bairro
                            FROM tweets t, bairro b
                            WHERE ST_Contains (b.the_geom, t.the_geom)
                                AND  t.id = f.id
                          );
```

DDL 3

Criação de tabelas antes da mineração de dados:

```
CREATE TABLE cluster_dbscan (
    id serial NOT NULL,
    descricao character varying(255),
    CONSTRAINT cluster_dbscan_pkey PRIMARY KEY (id),
    CONSTRAINT constraint_descricao UNIQUE (descricao)
);

CREATE TABLE rel_cluster_tweet (
    id serial NOT NULL,
    id_tab_cluster integer,
    id_tab_tweets integer,
    key_cluster integer,
    number_cluster integer,
    CONSTRAINT rel_cluster_tweet_pkey PRIMARY KEY (id),
    CONSTRAINT rel_cluster_tweet_id_tab_cluster_fkey FOREIGN KEY
        (id_tab_cluster)
        REFERENCES cluster_dbscan (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT rel_cluster_tweet_id_tab_tweets_fkey FOREIGN KEY
        (id_tab_tweets)
        REFERENCES sc_floripa_mais10_tweetsbrasil (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

DML 3

Deletar quem *tweetou* menos de 10 vezes:

```
DELETE FROM sc_floripa_mais10_tweetsbrasil
   WHERE
      user_id in (SELECT user_id
                  FROM sc_floripa_mais10_tweetsbrasil
                  GROUP BY user_id
                  HAVING count(*) < 10
                  );
```

DML 4

Para a extração da informação do dia da semana, foi utilizada a *query*:

```
   UPDATE
      tweets set dia_semana =
            case when extract(DOW FROM datetime) = 0
                  then 'domingo'
            when extract(DOW FROM datetime) = 1
                  then 'segunda'
            when extract(DOW FROM datetime) = 2
                  then 'terça'
            when extract(DOW FROM datetime) = 3
                  then 'quarta'
            when extract(DOW FROM datetime) = 4
                  then 'quinta'
            when extract(DOW FROM datetime) = 5
                  then 'sexta'
            when extract(DOW FROM datetime) = 6
                  then 'sábado'
            end;
```

DML 5

Para a extração da informação do período, foi utilizada a *query*:

```
UPDATE
    tweets set periodo =
        case when extract(HOUR FROM datetime) between 0 and  5
            then 'madrugada'
        when extract(HOUR FROM datetime) between 6 and 11
            then 'manhã'
        when extract(HOUR FROM datetime) between 12 and 17
            then 'tarde'
        when extract(HOUR FROM datetime) between 18 and 23
            then 'noite'
        end;
```

DDL 4

Função para retirar acentos ortográficos:

```
CREATE OR REPLACE FUNCTION sem_acento(character)
    RETURNS text AS
    $$ SELECT
    translate($1,'áàâãäéèêëíìïóòôõöúùûüÁÀÂÃÄÉÈÊËÍÌÏÓÒÔÕÖÚÙÛÜçÇ','aaaaaeeeeiiiooooouuuuAAAAAEEEEIIIOOOOOUUUUcC');
    $$ LANGUAGE sql VOLATILE;
```

DML 6

Atualizar lista de stopwords com o arquivo modificado previamente e definir o dicionário mais adequado como padrão:

```
UPDATE
        pg_ts_dict set dictinitoption='stopwords = "portuguese" ' where dictname='simple';

SET default_text_search_config TO 'simple';
```

DDL 5

Acrescentar coluna "vectors" na tabela que contem os tweets para armazenar o texto indexado:

```
ALTER TABLE
        sc_floripa_mais10_tweetsbrasil add vectors tsvector;

CREATE INDEX
         my_full_text_index_sc_floripa_mais10
        ON sc_floripa_mais10_tweetsbrasil
        USING gist(vectors);

UPDATE
        sc_floripa_mais10_tweetsbrasil set vectors=to_tsvector(text);
```

DML 7

Consulta exemplo que retorna as 15 palavras mais frequentes do cluster 0 do bairro Trindade no período da manhã:

```
SELECT word
        FROM stat('select vectors from sc_floripa_mais10_tweetsbrasil t
                        join rel_cluster_tweet r on t.id = r.id_tab_tweets
                        where r.id_tab_cluster = 1 and r.number_cluster = 0')
        ORDER BY ndoc desc limit 15;
```

ANEXO B - LISTA *STOPWORDS*

a	da	é	grande	nessas	podiam	será	ultimas
à	daquele	e'	grandes	nesta	pois	serao	últimas
agora	daqueles	ela	g	nestas	por	serão	ultimo
ainda	das	elas	h	ninguém	porem	seu	último
alguem	de	ele	há	no	porém	seus	ultimos
alguém	dela	eles	i	nos	porque	si	últimos
algum	delas	em	isso	nós	posso	sido	u
alguma	dele	enquanto	isto	nossa	pouca	so	um
algumas	deles	entre	j	nossas	poucas	só	uma
alguns	depois	era	ja	nosso	pouco	sob	umas
ampla	dessa	essa	já	nossos	poucos	sobre	uns
amplas	dessas	essas	k	num	primeiro	sua	v
amplo	desse	esse	l	numa	primeiros	suas	vendo
amplos	desses	esses	la	nunca	propria	t	ver
ante	desta	esta	la	o	própria	talvez	vez
antes	destas	está	lá	os	proprias	tambem	vindo
ao	deste	estamos	lhe	ou	próprias	também	vir
aos	deste	estao	lhes	outra	proprio	tampouco	vos
apos	destes	estão	lo	outras	próprio	te	vós
após	deve	estas	m	outro	proprios	tem	x
aquela	devem	estava	mas	outros	próprios	tendo	y
aquelas	devendo	estavam	me	p	q	tenha	w
aquele	dever	estavamos	mesma	para	quais	ter	z
aqueles	devera	estávamos	mesmas	pela	qual	teu	
aquilo	deverá	este	mesmo	pelas	quando	teus	
as	deverao	estes	mesmos	pelo	quanto	ti	
ate	deverão	estou	meu	pelos	quantos	tido	
até	deveria	eu	meus	pequena	que	tinha	
atraves	deveriam	f	minha	pequenas	quem	tinham	
através	devia	fazendo	minhas	pequeno	r	toda	
b	deviam	fazer	muita	pequenos	s	todas	
c	disse	feita	muitas	per	sao	todavia	
cada	disso	feitas	muito	perante	são	todo	
coisa	disto	feito	muitos	pode	se	todos	
coisas	dito	feitos	n	pôde	seja	tu	
com	diz	foi	na	podendo	sejam	tua	
como	dizem	for	nas	poder	sem	tuas	
contra	do	foram	nem	poderia	sempre	tudo	
contudo	dos	fosse	nenhum	poderiam	sendo	ultima	
d	e	fossem	nessa	podia	sera	última	

APÊNDICE 1 – CÓDIGO FONTE

```
package weka.clusterers;

public enum BairroEnum {

	TRINDADE(0, " bairro = 'TRINDADE' "),
	RIBEIRAO_DA_ILHA(1, " bairro = 'RIBEIRÃO DA ILHA' "),
	CANASVIEIRAS(2, " bairro = 'CANASVIEIRAS' "),
	CENTRO(3, " bairro = 'CENTRO' "),
	SANTO_ANTONIO_DE_LISBOA(4, " bairro = 'SANTO ANTÔNIO DE LISBOA' "),
	LAGOA_DA_CONCEICAO(5, " bairro = 'LAGOA DA CONCEIÇÃO' "),
	CACHOEIRA_DO_BOM_JESUS(6, " bairro = 'CACHOEIRA DO BOM JESUS' "),
	PANTANO_DO_SUL(7, " bairro = 'PÂNTANO DO SUL' "),
	CAMPECHE(8, " bairro = 'CAMPECHE' "),
	RATONES(9, " bairro = 'RATONES' "),
	SAO_JOAO_DO_RIO_VERMELHO(10, " bairro = 'SÃO JOÃO DO RIO VERMELHO' "),
	INGLESES_DO_RIO_VERMELHO(11, " bairro = 'INGLESES DO RIO VERMELHO' "),
	ITACORUBI(12, " bairro = 'ITACORUBI' "),
	SACO_GRANDE(13, " bairro = 'SACO GRANDE' "),
	CORREGO_GRANDE(14, " bairro = 'CÓRREGO GRANDE' "),
	BARRA_DA_LAGOA(15, " bairro = 'BARRA DA LAGOA' "),
	COSTEIRA_DO_PIRAJUBAE(16, " bairro = 'COSTEIRA DO PIRAJUBAÉ' "),
	MONTE_VERDE(17, " bairro = 'MONTE VERDE' "),
	SACO_DOS_LIMOES(18, " bairro = 'SACO DOS LIMÕES' "),
	CAPOEIRAS(19, " bairro = 'CAPOEIRAS' "),
	JOAO_PAULO(20, " bairro = 'JOÃO PAULO' "),
	PANTANAL(21, " bairro = 'PANTANAL' "),
	AGRONOMICA(22, " bairro = 'AGRONÔMICA' "),
	JARDIM_ATLANTICO(23, " bairro = 'JARDIM ATLÂNTICO' "),
	COQUEIROS(24, " bairro = 'COQUEIROS' "),
	ESTREITO(25, " bairro = 'ESTREITO' "),
	ABRAAO(26, " bairro = 'ABRAÃO' "),
	CANTO(27, " bairro = 'CANTO' "),
	BALNEARIO(28, " bairro = 'BALNEÁRIO' "),
	MONTE_CRISTO(29, " bairro = 'MONTE CRISTO' "),
	SANTA_MONICA(30, " bairro = 'SANTA MÔNICA' "),
	JOSE_MENDES(31, " bairro = 'JOSÉ MENDES' "),
	COLONINHA(32, " bairro = 'COLONINHA' "),
	ITAGUACU(33, " bairro = 'ITAGUAÇU' "),
	BOM_ABRIGO(34, " bairro = 'BOM ABRIGO' ");


	private Integer identificador;

	private String clausulaBairro;

	private BairroEnum (Integer id, String clausulaBairro) {
		this.identificador = id;
		this.clausulaBairro = clausulaBairro;
	}

	public Integer getIdentificador() {
		return identificador;
	}

	public String getClausulaBairro() {
		return clausulaBairro;
```

```
	}

}

package weka.clusterers;

import java.util.Vector;

import weka.core.Instance;

public class Cluster {

	private Vector<Instance> instances = new Vector<Instance>();

	private Double[] latLong = new Double[2];

	private String idTabCluster, index, color;

	public Cluster(Vector<Instance> instances) {
		this.instances = instances;
	}

	public Vector<Instance> getInstances() {
		return instances;
	}

	public void setInstances(Vector<Instance> instances) {
		this.instances = instances;
	}

	// se lat, entao latLong = 0; se long, entao latLong = 1
	public double avgLatLong(int latLong) {
		double avg = 0;
		for (int i = 0; i < instances.size(); i++) {
			avg += instances.get(i).value(latLong);
		}
		avg = avg / instances.size();
		this.latLong[latLong] = avg;
		return avg;
	}

	public Double[] getLatLong() {
		return latLong;
	}

	public void setLatLong(Double[] latLong) {
		this.latLong = latLong;
	}

	public String getIndex() {
		return index;
	}

	public void setIndex(String index) {
		this.index = index;
	}

	public String getColor() {
		return color;
```

```
	}

	public void setColor(String color) {
		this.color = color;
	}

	public String getIdTabCluster() {
		return idTabCluster;
	}

	public void setIdTabCluster(String idTabCluster) {
		this.idTabCluster = idTabCluster;

	}

}

package weka.clusterers;

import java.util.Vector;

public class ClusterList {

	public static String LAST_ID_CONSULTA = "1";
	public static String LAST_COLOR_CONSULTA;

	public static Vector<Cluster> CLUSTER_LIST = new Vector<Cluster>();

}

package weka.clusterers;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.LinkedList;
import java.util.List;
import java.util.Vector;

public class GoogleMapHtml {

	public GoogleMapHtml() {

		Vector<Cluster> clusters = ClusterList.CLUSTER_LIST;

		double avgLat = 0;
		double avgLong = 0;

		for (int i = 0; i < clusters.size(); i++) {
			avgLat += clusters.get(i).avgLatLong(0);
			avgLong += clusters.get(i).avgLatLong(1);
		}

		avgLat = avgLat / clusters.size();
		avgLong = avgLong / clusters.size();

		CENTRAL_POINT = CENTRAL_POINT.replace(LATIDUDE_CENTRAL_TO_REPLACE,
```

```
                        String.valueOf(avgLat));
                CENTRAL_POINT = CENTRAL_POINT.replace(LONGITUDE_CENTRAL_TO_REPLACE,
                        String.valueOf(avgLong));

                int nCluster = clusters.size();

                String clusterPointFinal = "";

                for (int i = 0; i < nCluster; i++) {
                        String clusterPoint = POINT_TO_CLUSTER;
                        clusterPoint = clusterPoint.replaceAll(COLOR_TO_REPLACE,
clusters
                                        .get(i).getColor());
                        clusterPoint = clusterPoint.replaceAll(LATIDUDE_TO_REPLACE,
                                        String.valueOf(clusters.get(i).avgLatLong(0)));
                        clusterPoint = clusterPoint.replaceAll(LONGITUDE_TO_REPLACE,
                                        String.valueOf(clusters.get(i).avgLatLong(1)));
                        clusterPoint = clusterPoint.replaceAll(INDEX_TO_REPLACE,
                                        String.valueOf(i));
                        clusterPoint = clusterPoint.replaceAll(INDEX_PLUS_TO_REPLACE,
                                        clusters.get(i).getIndex());

                        int indice = Integer.valueOf(clusters.get(i).getIndex());
                        indice = indice -1;

                        int idTabCluster =
Integer.valueOf(clusters.get(i).getIdTabCluster());
                        idTabCluster = idTabCluster-1;

                        List<String> p =
palavrasMaisFrequentesPorCluster(String.valueOf(idTabCluster),
String.valueOf(indice));

                        if (p != null) {

                                String palavras = "";

                                for (String string : p) {
                                        palavras += string + "<BR>";
                                }

                                clusterPoint = clusterPoint.replaceAll(
                                                PALAVRAS_CHAVE_TO_REPLACE, palavras);

                        }
                        clusterPointFinal += clusterPoint;
                }

                GENERAL_PAGE = GENERAL_PAGE.replace(CENTRAL_POINT_TO_REPLACE,
                                CENTRAL_POINT);
                GENERAL_PAGE = GENERAL_PAGE
                                .replace(POINT_TO_REPLACE, clusterPointFinal);

        }

        private final String POINT_TO_REPLACE = ":pontos_to_replace";
        private final String COLOR_TO_REPLACE = ":color_to_replace";
        private final String CENTRAL_POINT_TO_REPLACE = ":ponto_central_to_replace";
```

```
	private final String LATIDUDE_CENTRAL_TO_REPLACE =
":latitude_central_to_replace";
	private final String LONGITUDE_CENTRAL_TO_REPLACE =
":longitude_central_to_replace";
	private final String LATIDUDE_TO_REPLACE = ":latitude_to_replace";
	private final String LONGITUDE_TO_REPLACE = ":longitude_to_replace";
	private final String PALAVRAS_CHAVE_TO_REPLACE =
":palavras_chave_to_replace";
	private final String INDEX_TO_REPLACE = ":index_to_replace";
	private final String INDEX_PLUS_TO_REPLACE = ":index_plus_to_replace";

	private String CENTRAL_POINT = ":latitude_central_to_replace,
:longitude_central_to_replace";

	private String POINT_TO_CLUSTER = "pontosLt[:index_to_replace] =
:latitude_to_replace;"
			+ "pontosLg[:index_to_replace] = :longitude_to_replace;"
			+ "html[:index_to_replace] = \" :palavras_chave_to_replace \";"
			+ "icon[:index_to_replace] = new GIcon(G_DEFAULT_ICON);"
			+ "icon[:index_to_replace].image = \"http://gmaps-
samples.googlecode.com/svn/trunk/:color_to_replace.png\";";

	private String GENERAL_PAGE =
"<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\"
\"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">"
			+ "<html xmlns=\"http://www.w3.org/1999/xhtml\">"
			+ "<head>"
			+ "<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=utf-8\" />"
			+ "<title>TCC - Renata de Jesus Silva</title>"
			+ "<script
src=\"http://maps.google.com/maps?file=api&amp;v=2&amp;key=ABQIAAAAUcHQ2ldBNAAr0jS
88qEedBSItM6FQrLOEZkwRIcbYFrPk3yJGBSa_-BBB2_i0-BidwMJgkf4qz0ucw&amp;hl=pt\"
type=\"text/javascript\"></script>"
			+ "            "
			+ "   <script type=\"text/javascript\">"
			+ ""
			+ "    var map = null;"
			+ ""
			+ "    function initialize() {"
			+ "      if (GBrowserIsCompatible()) {"
			+ "        var map = new
GMap2(document.getElementById(\"mapa\"));"
			+ "         map.setCenter(new
GLatLng(:ponto_central_to_replace), 11);"
			+ "                var ui = new GMapUIOptions();"
			+ "          ui.maptypes = {normal:true};"
			+ "          ui.zoom = {doubleclick:true, scrollwheel:true};"
			+ "          ui.controls = {largemapcontrol3d:true,
scalecontrol:true, draggable: false} ;"
			+ "          ui.keyboard = false;"
			+ "          map.setUI(ui);"
			+ "          map.addControl(new GOverviewMapControl(new
GSize(100,100)));"
			+ "          var pontosLg = new Array();"
			+ "          var pontosLt = new Array();"
			+ "          var html = new Array();"
			+ "          var icon = new Array();"
			+ "          :pontos_to_replace"
```

```
                    + "          for (var i = 0; i < pontosLg.length; i++) {"
                    + "                map.addOverlay(criarMarca(pontosLt[i],
pontosLg[i], html[i],icon[i]));"
                    + "        }"
                    + "      }"
                    + "    }"
                    + "    function criarMarca(lat, lng, html, icon){"
                    + "          var point = new GLatLng(lat,lng);"
                    + "          var marca = new GMarker(point,{icon:icon,
draggable:false});"
                    + "          if(html != null){"
                    + "                GEvent.addListener(marca, \"click\",
function() {"
                    + "                marca.openInfoWindowHtml(html);"
                    + "                });"
                    + "          }"
                    + "    return marca;"
                    + "    }"
                    + "</script>"
                    + "</head>"
                    + "<body onload=\"initialize()\" onunload=\"GUnload()\">"
                    + "<h1>Analise de Dados Espaco-temporais Gerados por
Dispositivos Moveis na Rede Social Twitter</h1>"
                    + "<div id=\"mapa\" style=\"height:480px;width:100%;border:1px
solid #CCC;\">"
                    + "</div>"
                    + "<table>"
                    + "<tr style=\"height:50px;width: 500px;\"> "
                    + "<td style=\"width: 200px; font-weight: bold;\">Segunda a
Sexta:</td>"
                    + "<td style=\"width: 200px;\"> manha:"
                    + "    <img src=\"http://gmaps-
samples.googlecode.com/svn/trunk/markers/orange/blank.png\">"
                    + "</td>"
                    + "<td style=\"width: 200px;\"> tarde:"
                    + "    <img src=\"http://gmaps-
samples.googlecode.com/svn/trunk/markers/blue/blank.png\">"
                    + "</td>"
                    + "<td style=\"width: 200px;\"> noite:"
                    + "    <img src=\"http://gmaps-
samples.googlecode.com/svn/trunk/markers/red/blank.png\">"
                    + "</td>"
                    + "</tr>"
                    + "<tr style=\"width: 500px;\">"
                    + "<td style=\"width: 200px; font-weight: bold;\">Sabado e
Domingo:</td>"
                    + "<td style=\"width: 200px;\"> manha:"
                    + "    <img src=\"http://gmaps-
samples.googlecode.com/svn/trunk/slides/images/marker-yellow.png\">"
                    + "</td>"
                    + "<td style=\"width: 200px;\"> tarde:"
                    + "    <img src=\"http://gmaps-
samples.googlecode.com/svn/trunk/slides/images/blue-dot.png\">"
                    + "</td>"
                    + "<td style=\"width: 200px;\"> noite:"
                    + "    <img src=\"http://gmaps-
samples.googlecode.com/svn/trunk/slides/images/marker.png\">"
                    + "</td>" + "</tr>" + "</table>" + "</body>" + "</html>";
```

```
	public String getMapaHtml() {
		return this.GENERAL_PAGE;
	}

	public List<String> palavrasMaisFrequentesPorCluster(String id_tab_cluster,
			String number_cluster) {
		List<String> palavras = null;

		Connection connection = null;
		try {
			String driverName = "org.postgresql.Driver";


			Class.forName(driverName);


			String url = "jdbc:postgresql://localhost:5432/twitter";


			String username = "postgres";
			String password = "postgres";
			connection = DriverManager.getConnection(url, username,
password);

			Statement stmt = connection.createStatement();

			String sqlDescricao = "select descricao from cluster_dbscan
where id = " + id_tab_cluster;

			ResultSet rs = stmt.executeQuery(sqlDescricao);

			palavras = new LinkedList();

			while (rs.next()) {
				String str = rs.getString("descricao");
				str = str.substring(0, str.indexOf('_'));
				palavras.add(str);
				palavras.add("Cluster: " +number_cluster);
				palavras.add("");
			}

			String sql = "select word from stat('select vectors from
sc_floripa_mais10_tweetsbrasil t "
					+ "join rel_cluster_tweet r on t.id =
r.id_tab_tweets "
					+ "where r.id_tab_cluster = "
					+ id_tab_cluster
					+ " and r.number_cluster = "
					+ number_cluster
					+ " ') order by ndoc desc limit 15;";

			rs = stmt.executeQuery(sql);

			while (rs.next()) {
				String str = rs.getString("word");

				palavras.add(str);

			}
```

```
                        stmt.close();
                        connection.close();

                } catch (ClassNotFoundException e) {
                        e.printStackTrace();
                } catch (SQLException e) {
                        e.printStackTrace();
                }

                return palavras;
        }

}

package weka.clusterers;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;

public class MapSaver {

        public static void save(String conteudo) {

                gravarArquivo(conteudo);

                try {
                        URI uri;
                        uri = new URI("mapa.html");
                        uri.normalize();
                } catch (URISyntaxException e) {
                        e.printStackTrace();
                }
        }

        public static void gravarArquivo(String conteudo) {
          File arquivo;

          arquivo = new File("mapa.html");
          FileOutputStream fos;
                try {
                        fos = new FileOutputStream(arquivo);
                        fos.write(conteudo.getBytes());
                        fos.close();
                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }

}

package weka.clusterers;
```

```
public enum PeriodoEnum {

	MANHA(0, " periodo = 'manhã' "),
	TARDE(1, " periodo = 'tarde' "),
	NOITE(2, " (periodo = 'noite' or periodo = 'madrugada' ) ");

	private Integer identificador;

	private String clausulaPeriodo;

	private PeriodoEnum (Integer id, String clausulaPeriodo) {
		this.identificador = id;
		this.clausulaPeriodo = clausulaPeriodo;
	}

	public Integer getIdentificador() {
		return identificador;
	}

	public String getClausulaPeriodo() {
		return clausulaPeriodo;
	}

}

package weka.clusterers;

public enum TabClusterEnum {

	TRINDADE_MANHA_DIASEMANA(1, BairroEnum.TRINDADE, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
	TRINDADE_TARDE_DIASEMANA(2, BairroEnum.TRINDADE, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
	TRINDADE_NOITE_DIASEMANA(3, BairroEnum.TRINDADE, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
	TRINDADE_MANHA_FIMSEMANA(4, BairroEnum.TRINDADE, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
	TRINDADE_TARDE_FIMSEMANA(5, BairroEnum.TRINDADE, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
	TRINDADE_NOITE_FIMSEMANA(6, BairroEnum.TRINDADE, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

	RIBEIRAO_DA_ILHA_MANHA_DIASEMANA(7, BairroEnum.RIBEIRAO_DA_ILHA,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
	RIBEIRAO_DA_ILHA_TARDE_DIASEMANA(8, BairroEnum.RIBEIRAO_DA_ILHA,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
	RIBEIRAO_DA_ILHA_NOITE_DIASEMANA(9, BairroEnum.RIBEIRAO_DA_ILHA,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
	RIBEIRAO_DA_ILHA_MANHA_FIMSEMANA(10, BairroEnum.RIBEIRAO_DA_ILHA,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
	RIBEIRAO_DA_ILHA_TARDE_FIMSEMANA(11, BairroEnum.RIBEIRAO_DA_ILHA,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
	RIBEIRAO_DA_ILHA_NOITE_FIMSEMANA(12, BairroEnum.RIBEIRAO_DA_ILHA,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

	CANASVIEIRAS_MANHA_DIASEMANA(13, BairroEnum.CANASVIEIRAS, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
	CANASVIEIRAS_TARDE_DIASEMANA(14, BairroEnum.CANASVIEIRAS, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
```

```
      CANASVIEIRAS_NOITE_DIASEMANA(15, BairroEnum.CANASVIEIRAS, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      CANASVIEIRAS_MANHA_FIMSEMANA(16, BairroEnum.CANASVIEIRAS, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      CANASVIEIRAS_TARDE_FIMSEMANA(17, BairroEnum.CANASVIEIRAS, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      CANASVIEIRAS_NOITE_FIMSEMANA(18, BairroEnum.CANASVIEIRAS, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      CENTRO_MANHA_DIASEMANA(19, BairroEnum.CENTRO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      CENTRO_TARDE_DIASEMANA(20, BairroEnum.CENTRO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      CENTRO_NOITE_DIASEMANA(21, BairroEnum.CENTRO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      CENTRO_MANHA_FIMSEMANA(22, BairroEnum.CENTRO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      CENTRO_TARDE_FIMSEMANA(23, BairroEnum.CENTRO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      CENTRO_NOITE_FIMSEMANA(24, BairroEnum.CENTRO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      SANTO_ANTONIO_DE_LISBOA_MANHA_DIASEMANA(25,
BairroEnum.SANTO_ANTONIO_DE_LISBOA, PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA,
"markers/orange/blank"),
      SANTO_ANTONIO_DE_LISBOA_TARDE_DIASEMANA(26,
BairroEnum.SANTO_ANTONIO_DE_LISBOA, PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA,
"markers/blue/blank"),
      SANTO_ANTONIO_DE_LISBOA_NOITE_DIASEMANA(27,
BairroEnum.SANTO_ANTONIO_DE_LISBOA, PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA,
"markers/red/blank"),
      SANTO_ANTONIO_DE_LISBOA_MANHA_FIMSEMANA(28,
BairroEnum.SANTO_ANTONIO_DE_LISBOA, PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker-yellow"),
      SANTO_ANTONIO_DE_LISBOA_TARDE_FIMSEMANA(29,
BairroEnum.SANTO_ANTONIO_DE_LISBOA, PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/blue-dot"),
      SANTO_ANTONIO_DE_LISBOA_NOITE_FIMSEMANA(30,
BairroEnum.SANTO_ANTONIO_DE_LISBOA, PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker"),

      LAGOA_DA_CONCEICAO_MANHA_DIASEMANA(31, BairroEnum.LAGOA_DA_CONCEICAO,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      LAGOA_DA_CONCEICAO_TARDE_DIASEMANA(32, BairroEnum.LAGOA_DA_CONCEICAO,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      LAGOA_DA_CONCEICAO_NOITE_DIASEMANA(33, BairroEnum.LAGOA_DA_CONCEICAO,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      LAGOA_DA_CONCEICAO_MANHA_FIMSEMANA(34, BairroEnum.LAGOA_DA_CONCEICAO,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      LAGOA_DA_CONCEICAO_TARDE_FIMSEMANA(35, BairroEnum.LAGOA_DA_CONCEICAO,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      LAGOA_DA_CONCEICAO_NOITE_FIMSEMANA(36, BairroEnum.LAGOA_DA_CONCEICAO,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      CORREGO_GRANDE_MANHA_DIASEMANA(37, BairroEnum.CORREGO_GRANDE,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      CORREGO_GRANDE_TARDE_DIASEMANA(38, BairroEnum.CORREGO_GRANDE,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
```

```
      CORREGO_GRANDE_NOITE_DIASEMANA(39, BairroEnum.CORREGO_GRANDE,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      CORREGO_GRANDE_MANHA_FIMSEMANA(40, BairroEnum.CORREGO_GRANDE,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      CORREGO_GRANDE_TARDE_FIMSEMANA(41, BairroEnum.CORREGO_GRANDE,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      CORREGO_GRANDE_NOITE_FIMSEMANA(42, BairroEnum.CORREGO_GRANDE,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      SANTA_MONICA_MANHA_DIASEMANA(43, BairroEnum.SANTA_MONICA, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      SANTA_MONICA_TARDE_DIASEMANA(44, BairroEnum.SANTA_MONICA, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      SANTA_MONICA_NOITE_DIASEMANA(45, BairroEnum.SANTA_MONICA, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      SANTA_MONICA_MANHA_FIMSEMANA(46, BairroEnum.SANTA_MONICA, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      SANTA_MONICA_TARDE_FIMSEMANA(47, BairroEnum.SANTA_MONICA, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      SANTA_MONICA_NOITE_FIMSEMANA(48, BairroEnum.SANTA_MONICA, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      CAMPECHE_MANHA_DIASEMANA(49, BairroEnum.CAMPECHE, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      CAMPECHE_TARDE_DIASEMANA(50, BairroEnum.CAMPECHE, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      CAMPECHE_NOITE_DIASEMANA(51, BairroEnum.CAMPECHE, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      CAMPECHE_MANHA_FIMSEMANA(52, BairroEnum.CAMPECHE, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      CAMPECHE_TARDE_FIMSEMANA(53, BairroEnum.CAMPECHE, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      CAMPECHE_NOITE_FIMSEMANA(54, BairroEnum.CAMPECHE, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      COQUEIROS_MANHA_DIASEMANA(55, BairroEnum.COQUEIROS, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      COQUEIROS_TARDE_DIASEMANA(56, BairroEnum.COQUEIROS, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      COQUEIROS_NOITE_DIASEMANA(57, BairroEnum.COQUEIROS, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      COQUEIROS_MANHA_FIMSEMANA(58, BairroEnum.COQUEIROS, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      COQUEIROS_TARDE_FIMSEMANA(59, BairroEnum.COQUEIROS, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      COQUEIROS_NOITE_FIMSEMANA(60, BairroEnum.COQUEIROS, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      ESTREITO_MANHA_DIASEMANA(61, BairroEnum.ESTREITO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      ESTREITO_TARDE_DIASEMANA(62, BairroEnum.ESTREITO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      ESTREITO_NOITE_DIASEMANA(63, BairroEnum.ESTREITO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      ESTREITO_MANHA_FIMSEMANA(64, BairroEnum.ESTREITO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      ESTREITO_TARDE_FIMSEMANA(65, BairroEnum.ESTREITO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
```

```
        ESTREITO_NOITE_FIMSEMANA(66, BairroEnum.ESTREITO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        INGLESES_DO_RIO_VERMELHO_MANHA_DIASEMANA(67,
BairroEnum.INGLESES_DO_RIO_VERMELHO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        INGLESES_DO_RIO_VERMELHO_TARDE_DIASEMANA(68,
BairroEnum.INGLESES_DO_RIO_VERMELHO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        INGLESES_DO_RIO_VERMELHO_NOITE_DIASEMANA(69,
BairroEnum.INGLESES_DO_RIO_VERMELHO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        INGLESES_DO_RIO_VERMELHO_MANHA_FIMSEMANA(70,
BairroEnum.INGLESES_DO_RIO_VERMELHO, PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker-yellow"),
        INGLESES_DO_RIO_VERMELHO_TARDE_FIMSEMANA(71,
BairroEnum.INGLESES_DO_RIO_VERMELHO, PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/blue-dot"),
        INGLESES_DO_RIO_VERMELHO_NOITE_FIMSEMANA(72,
BairroEnum.INGLESES_DO_RIO_VERMELHO, PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker"),

        ITACORUBI_MANHA_DIASEMANA(73, BairroEnum.ITACORUBI, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        ITACORUBI_TARDE_DIASEMANA(74, BairroEnum.ITACORUBI, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        ITACORUBI_NOITE_DIASEMANA(75, BairroEnum.ITACORUBI, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        ITACORUBI_MANHA_FIMSEMANA(76, BairroEnum.ITACORUBI, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        ITACORUBI_TARDE_FIMSEMANA(77, BairroEnum.ITACORUBI, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        ITACORUBI_NOITE_FIMSEMANA(78, BairroEnum.ITACORUBI, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        SACO_GRANDE_MANHA_DIASEMANA(79, BairroEnum.SACO_GRANDE, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        SACO_GRANDE_TARDE_DIASEMANA(80, BairroEnum.SACO_GRANDE, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        SACO_GRANDE_NOITE_DIASEMANA(81, BairroEnum.SACO_GRANDE, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        SACO_GRANDE_MANHA_FIMSEMANA(82, BairroEnum.SACO_GRANDE, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        SACO_GRANDE_TARDE_FIMSEMANA(83, BairroEnum.SACO_GRANDE, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        SACO_GRANDE_NOITE_FIMSEMANA(84, BairroEnum.SACO_GRANDE, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        MONTE_VERDE_MANHA_DIASEMANA(85, BairroEnum.MONTE_VERDE, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        MONTE_VERDE_TARDE_DIASEMANA(86, BairroEnum.MONTE_VERDE, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        MONTE_VERDE_NOITE_DIASEMANA(87, BairroEnum.MONTE_VERDE, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        MONTE_VERDE_MANHA_FIMSEMANA(88, BairroEnum.MONTE_VERDE, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        MONTE_VERDE_TARDE_FIMSEMANA(89, BairroEnum.MONTE_VERDE, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
```

```
        MONTE_VERDE_NOITE_FIMSEMANA(90, BairroEnum.MONTE_VERDE, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        SACO_DOS_LIMOES_MANHA_DIASEMANA(91, BairroEnum.SACO_DOS_LIMOES,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        SACO_DOS_LIMOES_TARDE_DIASEMANA(92, BairroEnum.SACO_DOS_LIMOES,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        SACO_DOS_LIMOES_NOITE_DIASEMANA(93, BairroEnum.SACO_DOS_LIMOES,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        SACO_DOS_LIMOES_MANHA_FIMSEMANA(94, BairroEnum.SACO_DOS_LIMOES,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        SACO_DOS_LIMOES_TARDE_FIMSEMANA(95, BairroEnum.SACO_DOS_LIMOES,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        SACO_DOS_LIMOES_NOITE_FIMSEMANA(96, BairroEnum.SACO_DOS_LIMOES,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        CANTO_MANHA_DIASEMANA(97, BairroEnum.CANTO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        CANTO_TARDE_DIASEMANA(98, BairroEnum.CANTO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        CANTO_NOITE_DIASEMANA(99, BairroEnum.CANTO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        CANTO_MANHA_FIMSEMANA(100, BairroEnum.CANTO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        CANTO_TARDE_FIMSEMANA(101, BairroEnum.CANTO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        CANTO_NOITE_FIMSEMANA(102, BairroEnum.CANTO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        JOAO_PAULO_MANHA_DIASEMANA(103, BairroEnum.JOAO_PAULO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        JOAO_PAULO_TARDE_DIASEMANA(104, BairroEnum.JOAO_PAULO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        JOAO_PAULO_NOITE_DIASEMANA(105, BairroEnum.JOAO_PAULO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        JOAO_PAULO_MANHA_FIMSEMANA(106, BairroEnum.JOAO_PAULO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        JOAO_PAULO_TARDE_FIMSEMANA(107, BairroEnum.JOAO_PAULO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        JOAO_PAULO_NOITE_FIMSEMANA(108, BairroEnum.JOAO_PAULO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        CAPOEIRAS_MANHA_DIASEMANA(109, BairroEnum.CAPOEIRAS, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        CAPOEIRAS_TARDE_DIASEMANA(110, BairroEnum.CAPOEIRAS, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        CAPOEIRAS_NOITE_DIASEMANA(111, BairroEnum.CAPOEIRAS, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        CAPOEIRAS_MANHA_FIMSEMANA(112, BairroEnum.CAPOEIRAS, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        CAPOEIRAS_TARDE_FIMSEMANA(113, BairroEnum.CAPOEIRAS, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        CAPOEIRAS_NOITE_FIMSEMANA(114, BairroEnum.CAPOEIRAS, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        AGRONOMICA_MANHA_DIASEMANA(115, BairroEnum.AGRONOMICA, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        AGRONOMICA_TARDE_DIASEMANA(116, BairroEnum.AGRONOMICA, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
```

```
      AGRONOMICA_NOITE_DIASEMANA(117, BairroEnum.AGRONOMICA, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      AGRONOMICA_MANHA_FIMSEMANA(118, BairroEnum.AGRONOMICA, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      AGRONOMICA_TARDE_FIMSEMANA(119, BairroEnum.AGRONOMICA, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      AGRONOMICA_NOITE_FIMSEMANA(120, BairroEnum.AGRONOMICA, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      PANTANAL_MANHA_DIASEMANA(121, BairroEnum.PANTANAL, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      PANTANAL_TARDE_DIASEMANA(122, BairroEnum.PANTANAL, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      PANTANAL_NOITE_DIASEMANA(123, BairroEnum.PANTANAL, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      PANTANAL_MANHA_FIMSEMANA(124, BairroEnum.PANTANAL, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      PANTANAL_TARDE_FIMSEMANA(125, BairroEnum.PANTANAL, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      PANTANAL_NOITE_FIMSEMANA(126, BairroEnum.PANTANAL, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      COSTEIRA_DO_PIRAJUBAE_MANHA_DIASEMANA(127, BairroEnum.COSTEIRA_DO_PIRAJUBAE,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      COSTEIRA_DO_PIRAJUBAE_TARDE_DIASEMANA(128, BairroEnum.COSTEIRA_DO_PIRAJUBAE,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      COSTEIRA_DO_PIRAJUBAE_NOITE_DIASEMANA(129, BairroEnum.COSTEIRA_DO_PIRAJUBAE,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      COSTEIRA_DO_PIRAJUBAE_MANHA_FIMSEMANA(130, BairroEnum.COSTEIRA_DO_PIRAJUBAE,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      COSTEIRA_DO_PIRAJUBAE_TARDE_FIMSEMANA(131, BairroEnum.COSTEIRA_DO_PIRAJUBAE,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      COSTEIRA_DO_PIRAJUBAE_NOITE_FIMSEMANA(132, BairroEnum.COSTEIRA_DO_PIRAJUBAE,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      JARDIM_ATLANTICO_MANHA_DIASEMANA(133, BairroEnum.JARDIM_ATLANTICO,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      JARDIM_ATLANTICO_TARDE_DIASEMANA(134, BairroEnum.JARDIM_ATLANTICO,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      JARDIM_ATLANTICO_NOITE_DIASEMANA(135, BairroEnum.JARDIM_ATLANTICO,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      JARDIM_ATLANTICO_MANHA_FIMSEMANA(136, BairroEnum.JARDIM_ATLANTICO,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      JARDIM_ATLANTICO_TARDE_FIMSEMANA(137, BairroEnum.JARDIM_ATLANTICO,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
      JARDIM_ATLANTICO_NOITE_FIMSEMANA(138, BairroEnum.JARDIM_ATLANTICO,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

      RATONES_MANHA_DIASEMANA(139, BairroEnum.RATONES, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
      RATONES_TARDE_DIASEMANA(140, BairroEnum.RATONES, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
      RATONES_NOITE_DIASEMANA(141, BairroEnum.RATONES, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
      RATONES_MANHA_FIMSEMANA(142, BairroEnum.RATONES, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
      RATONES_TARDE_FIMSEMANA(143, BairroEnum.RATONES, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
```

```
        RATONES_NOITE_FIMSEMANA(144, BairroEnum.RATONES, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        SAO_JOAO_DO_RIO_VERMELHO_MANHA_DIASEMANA(145,
BairroEnum.SAO_JOAO_DO_RIO_VERMELHO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        SAO_JOAO_DO_RIO_VERMELHO_TARDE_DIASEMANA(146,
BairroEnum.SAO_JOAO_DO_RIO_VERMELHO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        SAO_JOAO_DO_RIO_VERMELHO_NOITE_DIASEMANA(147,
BairroEnum.SAO_JOAO_DO_RIO_VERMELHO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        SAO_JOAO_DO_RIO_VERMELHO_MANHA_FIMSEMANA(148,
BairroEnum.SAO_JOAO_DO_RIO_VERMELHO, PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker-yellow"),
        SAO_JOAO_DO_RIO_VERMELHO_TARDE_FIMSEMANA(149,
BairroEnum.SAO_JOAO_DO_RIO_VERMELHO, PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/blue-dot"),
        SAO_JOAO_DO_RIO_VERMELHO_NOITE_FIMSEMANA(150,
BairroEnum.SAO_JOAO_DO_RIO_VERMELHO, PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker"),

        ABRAAO_MANHA_DIASEMANA(151, BairroEnum.ABRAAO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        ABRAAO_TARDE_DIASEMANA(152, BairroEnum.ABRAAO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        ABRAAO_NOITE_DIASEMANA(153, BairroEnum.ABRAAO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        ABRAAO_MANHA_FIMSEMANA(154, BairroEnum.ABRAAO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        ABRAAO_TARDE_FIMSEMANA(155, BairroEnum.ABRAAO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        ABRAAO_NOITE_FIMSEMANA(156, BairroEnum.ABRAAO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        BARRA_DA_LAGOA_MANHA_DIASEMANA(157, BairroEnum.BARRA_DA_LAGOA,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        BARRA_DA_LAGOA_TARDE_DIASEMANA(158, BairroEnum.BARRA_DA_LAGOA,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        BARRA_DA_LAGOA_NOITE_DIASEMANA(159, BairroEnum.BARRA_DA_LAGOA,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        BARRA_DA_LAGOA_MANHA_FIMSEMANA(160, BairroEnum.BARRA_DA_LAGOA,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        BARRA_DA_LAGOA_TARDE_FIMSEMANA(161, BairroEnum.BARRA_DA_LAGOA,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        BARRA_DA_LAGOA_NOITE_FIMSEMANA(162, BairroEnum.BARRA_DA_LAGOA,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        BALNEARIO_MANHA_DIASEMANA(163, BairroEnum.BALNEARIO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        BALNEARIO_TARDE_DIASEMANA(164, BairroEnum.BALNEARIO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        BALNEARIO_NOITE_DIASEMANA(165, BairroEnum.BALNEARIO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        BALNEARIO_MANHA_FIMSEMANA(166, BairroEnum.BALNEARIO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        BALNEARIO_TARDE_FIMSEMANA(167, BairroEnum.BALNEARIO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
```

```
        BALNEARIO_NOITE_FIMSEMANA(168, BairroEnum.BALNEARIO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        MONTE_CRISTO_MANHA_DIASEMANA(169, BairroEnum.MONTE_CRISTO,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        MONTE_CRISTO_TARDE_DIASEMANA(170, BairroEnum.MONTE_CRISTO,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        MONTE_CRISTO_NOITE_DIASEMANA(171, BairroEnum.MONTE_CRISTO,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        MONTE_CRISTO_MANHA_FIMSEMANA(172, BairroEnum.MONTE_CRISTO,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        MONTE_CRISTO_TARDE_FIMSEMANA(173, BairroEnum.MONTE_CRISTO,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        MONTE_CRISTO_NOITE_FIMSEMANA(174, BairroEnum.MONTE_CRISTO,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        PANTANO_DO_SUL_MANHA_DIASEMANA(175, BairroEnum.PANTANO_DO_SUL,
PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        PANTANO_DO_SUL_TARDE_DIASEMANA(176, BairroEnum.PANTANO_DO_SUL,
PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        PANTANO_DO_SUL_NOITE_DIASEMANA(177, BairroEnum.PANTANO_DO_SUL,
PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        PANTANO_DO_SUL_MANHA_FIMSEMANA(178, BairroEnum.PANTANO_DO_SUL,
PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        PANTANO_DO_SUL_TARDE_FIMSEMANA(179, BairroEnum.PANTANO_DO_SUL,
PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        PANTANO_DO_SUL_NOITE_FIMSEMANA(180, BairroEnum.PANTANO_DO_SUL,
PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        JOSE_MENDES_MANHA_DIASEMANA(181, BairroEnum.JOSE_MENDES, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        JOSE_MENDES_TARDE_DIASEMANA(182, BairroEnum.JOSE_MENDES, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        JOSE_MENDES_NOITE_DIASEMANA(183, BairroEnum.JOSE_MENDES, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        JOSE_MENDES_MANHA_FIMSEMANA(184, BairroEnum.JOSE_MENDES, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        JOSE_MENDES_TARDE_FIMSEMANA(185, BairroEnum.JOSE_MENDES, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        JOSE_MENDES_NOITE_FIMSEMANA(186, BairroEnum.JOSE_MENDES, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        COLONINHA_MANHA_DIASEMANA(187, BairroEnum.COLONINHA, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        COLONINHA_TARDE_DIASEMANA(188, BairroEnum.COLONINHA, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
        COLONINHA_NOITE_DIASEMANA(189, BairroEnum.COLONINHA, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
        COLONINHA_MANHA_FIMSEMANA(190, BairroEnum.COLONINHA, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
        COLONINHA_TARDE_FIMSEMANA(191, BairroEnum.COLONINHA, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
        COLONINHA_NOITE_FIMSEMANA(192, BairroEnum.COLONINHA, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

        ITAGUACU_MANHA_DIASEMANA(193, BairroEnum.ITAGUACU, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
        ITAGUACU_TARDE_DIASEMANA(194, BairroEnum.ITAGUACU, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
```

```
	ITAGUACU_NOITE_DIASEMANA(195, BairroEnum.ITAGUACU, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
	ITAGUACU_MANHA_FIMSEMANA(196, BairroEnum.ITAGUACU, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
	ITAGUACU_TARDE_FIMSEMANA(197, BairroEnum.ITAGUACU, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
	ITAGUACU_NOITE_FIMSEMANA(198, BairroEnum.ITAGUACU, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

	BOM_ABRIGO_MANHA_DIASEMANA(199, BairroEnum.BOM_ABRIGO, PeriodoEnum.MANHA,
TipoSemanaEnum.SEG_A_SEXTA, "markers/orange/blank"),
	BOM_ABRIGO_TARDE_DIASEMANA(200, BairroEnum.BOM_ABRIGO, PeriodoEnum.TARDE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/blue/blank"),
	BOM_ABRIGO_NOITE_DIASEMANA(201, BairroEnum.BOM_ABRIGO, PeriodoEnum.NOITE,
TipoSemanaEnum.SEG_A_SEXTA, "markers/red/blank"),
	BOM_ABRIGO_MANHA_FIMSEMANA(202, BairroEnum.BOM_ABRIGO, PeriodoEnum.MANHA,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker-yellow"),
	BOM_ABRIGO_TARDE_FIMSEMANA(203, BairroEnum.BOM_ABRIGO, PeriodoEnum.TARDE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/blue-dot"),
	BOM_ABRIGO_NOITE_FIMSEMANA(204, BairroEnum.BOM_ABRIGO, PeriodoEnum.NOITE,
TipoSemanaEnum.FIM_SEMANA, "slides/images/marker"),

	CACHOEIRA_DO_BOM_JESUS_MANHA_DIASEMANA(205,
BairroEnum.CACHOEIRA_DO_BOM_JESUS, PeriodoEnum.MANHA, TipoSemanaEnum.SEG_A_SEXTA,
"markers/orange/blank"),
	CACHOEIRA_DO_BOM_JESUS_TARDE_DIASEMANA(206,
BairroEnum.CACHOEIRA_DO_BOM_JESUS, PeriodoEnum.TARDE, TipoSemanaEnum.SEG_A_SEXTA,
"markers/blue/blank"),
	CACHOEIRA_DO_BOM_JESUS_NOITE_DIASEMANA(207,
BairroEnum.CACHOEIRA_DO_BOM_JESUS, PeriodoEnum.NOITE, TipoSemanaEnum.SEG_A_SEXTA,
"markers/red/blank"),
	CACHOEIRA_DO_BOM_JESUS_MANHA_FIMSEMANA(208,
BairroEnum.CACHOEIRA_DO_BOM_JESUS, PeriodoEnum.MANHA, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker-yellow"),
	CACHOEIRA_DO_BOM_JESUS_TARDE_FIMSEMANA(209,
BairroEnum.CACHOEIRA_DO_BOM_JESUS, PeriodoEnum.TARDE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/blue-dot"),
	CACHOEIRA_DO_BOM_JESUS_NOITE_FIMSEMANA(210,
BairroEnum.CACHOEIRA_DO_BOM_JESUS, PeriodoEnum.NOITE, TipoSemanaEnum.FIM_SEMANA,
"slides/images/marker");

	private Integer id;
	private BairroEnum bairro;
	private PeriodoEnum periodo;
	private TipoSemanaEnum tipoSemana;
	private String cor;

	private TabClusterEnum (Integer id, BairroEnum bairro, PeriodoEnum periodo,
TipoSemanaEnum tipoSemana, String cor) {
		this.id = id;
		this.bairro = bairro;
		this.periodo = periodo;
		this.tipoSemana = tipoSemana;
		this.cor = cor;
	}

	public Integer getId() {
		return id;
	}
```

```
public BairroEnum getBairro() {
	return bairro;
}

public PeriodoEnum getPeriodo() {
	return periodo;
}

public TipoSemanaEnum getTipoSemana() {
	return tipoSemana;
}

public static TabClusterEnum get(Integer idConsulta) {
	TabClusterEnum tabCluster = null;
	switch (idConsulta) {
	case 1:
		tabCluster = TRINDADE_MANHA_DIASEMANA;
		break;
	case 2:
		tabCluster = TRINDADE_TARDE_DIASEMANA;
		break;
	case 3:
		tabCluster = TRINDADE_NOITE_DIASEMANA;
		break;
	case 4:
		tabCluster = TRINDADE_MANHA_FIMSEMANA;
		break;
	case 5:
		tabCluster = TRINDADE_TARDE_FIMSEMANA;
		break;
	case 6:
		tabCluster = TRINDADE_NOITE_FIMSEMANA;
		break;

	case 7:
		tabCluster = RIBEIRAO_DA_ILHA_MANHA_DIASEMANA;
		break;
	case 8:
		tabCluster = RIBEIRAO_DA_ILHA_TARDE_DIASEMANA;
		break;
	case 9:
		tabCluster = RIBEIRAO_DA_ILHA_NOITE_DIASEMANA;
		break;
	case 10:
		tabCluster = RIBEIRAO_DA_ILHA_MANHA_FIMSEMANA;
		break;
	case 11:
		tabCluster = RIBEIRAO_DA_ILHA_TARDE_FIMSEMANA;
		break;
	case 12:
		tabCluster = RIBEIRAO_DA_ILHA_NOITE_FIMSEMANA;
		break;

	case 13:
		tabCluster = CANASVIEIRAS_MANHA_DIASEMANA;
		break;
	case 14:
		tabCluster = CANASVIEIRAS_TARDE_DIASEMANA;
```

```
        break;
case 15:
        tabCluster = CANASVIEIRAS_NOITE_DIASEMANA;
        break;
case 16:
        tabCluster = CANASVIEIRAS_MANHA_FIMSEMANA;
        break;
case 17:
        tabCluster = CANASVIEIRAS_TARDE_FIMSEMANA;
        break;
case 18:
        tabCluster = CANASVIEIRAS_NOITE_FIMSEMANA;
        break;

case 19:
        tabCluster = CENTRO_MANHA_DIASEMANA;
        break;
case 20:
        tabCluster = CENTRO_TARDE_DIASEMANA;
        break;
case 21:
        tabCluster = CENTRO_NOITE_DIASEMANA;
        break;
case 22:
        tabCluster = CENTRO_MANHA_FIMSEMANA;
        break;
case 23:
        tabCluster = CENTRO_TARDE_FIMSEMANA;
        break;
case 24:
        tabCluster = CENTRO_NOITE_FIMSEMANA;
        break;

case 25:
        tabCluster = SANTO_ANTONIO_DE_LISBOA_MANHA_DIASEMANA;
        break;
case 26:
        tabCluster = SANTO_ANTONIO_DE_LISBOA_TARDE_DIASEMANA;
        break;
case 27:
        tabCluster = SANTO_ANTONIO_DE_LISBOA_NOITE_DIASEMANA;
        break;
case 28:
        tabCluster = SANTO_ANTONIO_DE_LISBOA_MANHA_FIMSEMANA;
        break;
case 29:
        tabCluster = SANTO_ANTONIO_DE_LISBOA_TARDE_FIMSEMANA;
        break;
case 30:
        tabCluster = SANTO_ANTONIO_DE_LISBOA_NOITE_FIMSEMANA;
        break;

case 31:
        tabCluster = LAGOA_DA_CONCEICAO_MANHA_DIASEMANA;
        break;
case 32:
        tabCluster = LAGOA_DA_CONCEICAO_TARDE_DIASEMANA;
        break;
case 33:
```

```
            tabCluster = LAGOA_DA_CONCEICAO_NOITE_DIASEMANA;
            break;
        case 34:
            tabCluster = LAGOA_DA_CONCEICAO_MANHA_FIMSEMANA;
            break;
        case 35:
            tabCluster = LAGOA_DA_CONCEICAO_TARDE_FIMSEMANA;
            break;
        case 36:
            tabCluster = LAGOA_DA_CONCEICAO_NOITE_FIMSEMANA;
            break;

        case 37:
            tabCluster = CORREGO_GRANDE_MANHA_DIASEMANA;
            break;
        case 38:
            tabCluster = CORREGO_GRANDE_TARDE_DIASEMANA;
            break;
        case 39:
            tabCluster = CORREGO_GRANDE_NOITE_DIASEMANA;
            break;
        case 40:
            tabCluster = CORREGO_GRANDE_MANHA_FIMSEMANA;
            break;
        case 41:
            tabCluster = CORREGO_GRANDE_TARDE_FIMSEMANA;
            break;
        case 42:
            tabCluster = CORREGO_GRANDE_NOITE_FIMSEMANA;
            break;

        case 43:
            tabCluster = SANTA_MONICA_MANHA_DIASEMANA;
            break;
        case 44:
            tabCluster = SANTA_MONICA_TARDE_DIASEMANA;
            break;
        case 45:
            tabCluster = SANTA_MONICA_NOITE_DIASEMANA;
            break;
        case 46:
            tabCluster = SANTA_MONICA_MANHA_FIMSEMANA;
            break;
        case 47:
            tabCluster = SANTA_MONICA_TARDE_FIMSEMANA;
            break;
        case 48:
            tabCluster = SANTA_MONICA_NOITE_FIMSEMANA;
            break;

        case 49:
            tabCluster = CAMPECHE_MANHA_DIASEMANA;
            break;
        case 50:
            tabCluster = CAMPECHE_TARDE_DIASEMANA;
            break;
        case 51:
            tabCluster = CAMPECHE_NOITE_DIASEMANA;
            break;
```

```
case 52:
      tabCluster = CAMPECHE_MANHA_FIMSEMANA;
      break;
case 53:
      tabCluster = CAMPECHE_TARDE_FIMSEMANA;
      break;
case 54:
      tabCluster = CAMPECHE_NOITE_FIMSEMANA;
      break;

case 55:
      tabCluster = COQUEIROS_MANHA_DIASEMANA;
      break;
case 56:
      tabCluster = COQUEIROS_TARDE_DIASEMANA;
      break;
case 57:
      tabCluster = COQUEIROS_NOITE_DIASEMANA;
      break;
case 58:
      tabCluster = COQUEIROS_MANHA_FIMSEMANA;
      break;
case 59:
      tabCluster = COQUEIROS_TARDE_FIMSEMANA;
      break;
case 60:
      tabCluster = COQUEIROS_NOITE_FIMSEMANA;
      break;

case 61:
      tabCluster = ESTREITO_MANHA_DIASEMANA;
      break;
case 62:
      tabCluster = ESTREITO_TARDE_DIASEMANA;
      break;
case 63:
      tabCluster = ESTREITO_NOITE_DIASEMANA;
      break;
case 64:
      tabCluster = ESTREITO_MANHA_FIMSEMANA;
      break;
case 65:
      tabCluster = ESTREITO_TARDE_FIMSEMANA;
      break;
case 66:
      tabCluster = ESTREITO_NOITE_FIMSEMANA;
      break;

case 67:
      tabCluster = INGLESES_DO_RIO_VERMELHO_MANHA_DIASEMANA;
      break;
case 68:
      tabCluster = INGLESES_DO_RIO_VERMELHO_TARDE_DIASEMANA;
      break;
case 69:
      tabCluster = INGLESES_DO_RIO_VERMELHO_NOITE_DIASEMANA;
      break;
case 70:
      tabCluster = INGLESES_DO_RIO_VERMELHO_MANHA_FIMSEMANA;
```

```
        break;
    case 71:
        tabCluster = INGLESES_DO_RIO_VERMELHO_TARDE_FIMSEMANA;
        break;
    case 72:
        tabCluster = INGLESES_DO_RIO_VERMELHO_NOITE_FIMSEMANA;
        break;

    case 73:
        tabCluster = ITACORUBI_MANHA_DIASEMANA;
        break;
    case 74:
        tabCluster = ITACORUBI_TARDE_DIASEMANA;
        break;
    case 75:
        tabCluster = ITACORUBI_NOITE_DIASEMANA;
        break;
    case 76:
        tabCluster = ITACORUBI_MANHA_FIMSEMANA;
        break;
    case 77:
        tabCluster = ITACORUBI_TARDE_FIMSEMANA;
        break;
    case 78:
        tabCluster = ITACORUBI_NOITE_FIMSEMANA;
        break;

    case 79:
        tabCluster = SACO_GRANDE_MANHA_DIASEMANA;
        break;
    case 80:
        tabCluster = SACO_GRANDE_TARDE_DIASEMANA;
        break;
    case 81:
        tabCluster = SACO_GRANDE_NOITE_DIASEMANA;
        break;
    case 82:
        tabCluster = SACO_GRANDE_MANHA_FIMSEMANA;
        break;
    case 83:
        tabCluster = SACO_GRANDE_TARDE_FIMSEMANA;
        break;
    case 84:
        tabCluster = SACO_GRANDE_NOITE_FIMSEMANA;
        break;

    case 85:
        tabCluster = MONTE_VERDE_MANHA_DIASEMANA;
        break;
    case 86:
        tabCluster = MONTE_VERDE_TARDE_DIASEMANA;
        break;
    case 87:
        tabCluster = MONTE_VERDE_NOITE_DIASEMANA;
        break;
    case 88:
        tabCluster = MONTE_VERDE_MANHA_FIMSEMANA;
        break;
    case 89:
```

```
        tabCluster = MONTE_VERDE_TARDE_FIMSEMANA;
        break;
case 90:
        tabCluster = MONTE_VERDE_NOITE_FIMSEMANA;
        break;

case 91:
        tabCluster = SACO_DOS_LIMOES_MANHA_DIASEMANA;
        break;
case 92:
        tabCluster = SACO_DOS_LIMOES_TARDE_DIASEMANA;
        break;
case 93:
        tabCluster = SACO_DOS_LIMOES_NOITE_DIASEMANA;
        break;
case 94:
        tabCluster = SACO_DOS_LIMOES_MANHA_FIMSEMANA;
        break;
case 95:
        tabCluster = SACO_DOS_LIMOES_TARDE_FIMSEMANA;
        break;
case 96:
        tabCluster = SACO_DOS_LIMOES_NOITE_FIMSEMANA;
        break;

case 97:
        tabCluster = CANTO_MANHA_DIASEMANA;
        break;
case 98:
        tabCluster = CANTO_TARDE_DIASEMANA;
        break;
case 99:
        tabCluster = CANTO_NOITE_DIASEMANA;
        break;
case 100:
        tabCluster = CANTO_MANHA_FIMSEMANA;
        break;
case 101:
        tabCluster = CANTO_TARDE_FIMSEMANA;
        break;
case 102:
        tabCluster = CANTO_NOITE_FIMSEMANA;
        break;

case 103:
        tabCluster = JOAO_PAULO_MANHA_DIASEMANA;
        break;
case 104:
        tabCluster = JOAO_PAULO_TARDE_DIASEMANA;
        break;
case 105:
        tabCluster = JOAO_PAULO_NOITE_DIASEMANA;
        break;
case 106:
        tabCluster = JOAO_PAULO_MANHA_FIMSEMANA;
        break;
case 107:
        tabCluster = JOAO_PAULO_TARDE_FIMSEMANA;
        break;
```

```
                case 108:
                        tabCluster = JOAO_PAULO_NOITE_FIMSEMANA;
                        break;

                case 109:
                        tabCluster = CAPOEIRAS_MANHA_DIASEMANA;
                        break;
                case 110:
                        tabCluster = CAPOEIRAS_TARDE_DIASEMANA;
                        break;
                case 111:
                        tabCluster = CAPOEIRAS_NOITE_DIASEMANA;
                        break;
                case 112:
                        tabCluster = CAPOEIRAS_MANHA_FIMSEMANA;
                        break;
                case 113:
                        tabCluster = CAPOEIRAS_TARDE_FIMSEMANA;
                        break;
                case 114:
                        tabCluster = CAPOEIRAS_NOITE_FIMSEMANA;
                        break;

                case 115:
                        tabCluster = AGRONOMICA_MANHA_DIASEMANA;
                        break;
                case 116:
                        tabCluster = AGRONOMICA_TARDE_DIASEMANA;
                        break;
                case 117:
                        tabCluster = AGRONOMICA_NOITE_DIASEMANA;
                        break;
                case 118:
                        tabCluster = AGRONOMICA_MANHA_FIMSEMANA;
                        break;
                case 119:
                        tabCluster = AGRONOMICA_TARDE_FIMSEMANA;
                        break;
                case 120:
                        tabCluster = AGRONOMICA_NOITE_FIMSEMANA;
                        break;

                default:
                        break;
                }
                return tabCluster;
        }

        public String getCor() {
                return this.cor;
        }

}

package weka.clusterers;

public enum TipoSemanaEnum {
```

```
	SEG_A_SEXTA(0, " dia_semana = 'segunda' or dia_semana = 'terça' or
dia_semana = 'quarta' or dia_semana = 'quinta' or dia_semana = 'sexta' "),
	FIM_SEMANA(1, " dia_semana = 'sábado' or dia_semana = 'domingo' ");

	private Integer identificador;

	private String clausulaDiaSemana;

	private TipoSemanaEnum (Integer id, String clausulaDiaSemana) {
		this.identificador = id;
		this.clausulaDiaSemana = clausulaDiaSemana;
	}

	public Integer getIdentificador() {
		return identificador;
	}

	public String getClausulaDiaSemana() {
		return clausulaDiaSemana;
	}

}
```

APÊNDICE 2 – ARTIGO

Análise espaço-temporal de mensagens do Twitter

Renata de J. Silva[1], Luis Otavio Alvares[1]

[1]Depto. Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

Abstract. We live in an era in which access to the Internet becomes increasingly common. Social networks such as Twitter microblog, have recorded a significant increase of posted messages. Some of these messages have the geographical coordinates of the location where they were issued. This paper proposes the analysis of these posts considering the spatio-temporal aspects in order to obtain knowledge about users of Twitter. For this, we propose changes in the Weka data mining toolkit to obtain better results on twitter data analysis. Experiments were performed with real data obtaining good results.

Resumo. *Estamos vivenciando uma era em que o acesso à internet torna-se cada vez mais frequente. As redes sociais, como o microblog Twitter, tem registrado um aumento significativo de mensagens postadas. Parte destas mensagens possui as coordenadas geográficas do local de onde foram emitidas. Este artigo propõe a análise destas mensagens considerando os aspectos espaço-temporais de modo a obter conhecimento sobre os usuários do Twitter. Para isto, propõe adaptações na ferramenta de data mining Weka de forma a obter melhores resultados. Experimentos foram realizados com dados reais, com bons resultados.*

1. Introdução

Com a popularização das redes sociais na internet, a disseminação e o acesso à informação tornou-se muito mais ágil. Uma dessas redes, o Twitter, na verdade mais considerado um micro blog do que uma rede social, tem características particulares: suas mensagens são limitadas a 140 caracteres e usualmente são postadas de dispositivos móveis como celulares e *smartphones*; a maioria das mensagens reflete onde o usuário está, ou o que ele está fazendo ou sentindo naquele momento; para receber as postagens de um usuário (ser um seguidor) não há necessidade de concordancia deste usuário.

Com mais de 500 milhões de usuários [UOL Tecnologia 2012] e 500 milhões de mensagens por dia [Olhar Digital UOL 2012], o Twitter é uma fonte impressionante de informações. Entretanto, analisar milhões de dados publicados diariamente no Twitter é muito trabalhoso e inviável manualmente. Uma alternativa é aplicar técnicas de mineração de dados. Alguns estudos já abordam este problema, mas muito pouco existe que considere os aspectos espacial e temporal simultaneamente.

Este trabalho tem o foco em mineração de dados utilizando a base de dados do Twitter, com *tweets* – mensagens publicadas no Twitter – georreferenciados. São apresentadas adaptações na ferramenta Weka para que as análises de dados espaço-temporais dos *tweets* possam se tornar mais eficazes e eficientes. Mais especificamente, é abordada a técnica de formação de agrupamentos com o algoritmo DBSCAN, cuja saída é incrementada com uma

visualização na forma de mapas e a indicação das palavras mais frequentes nas mensagens de cada cluster, de modo a se ter uma ideia geral do conteúdo das mensagens.

O restante do artigo está organizado como segue: a seção 2 apresenta alguns trabalhos relacionados; a seção 3 apresenta o que é a ferramenta Weka; a seção 4 apresenta o que foi adaptado nesta ferramenta a fim de melhorar a capacidade de resposta às análises; na seção 5 são apresentados alguns experimentos realizados; e por fim a seção 6 expõe a conclusão e trabalhos futuros.

2. Trabalhos Relacionados

As redes sociais na internet são relativamente recentes e o volume de seus dados tem crescido exponencialmente nos últimos anos. A descoberta de conhecimento neste novo tipo de dado tem suscitado muito interesse e vários trabalhos tem abordado o tema. Entretanto, trabalhos considerando os aspectos espaço-temporais das mensagens postadas são bem menos numerosos. Por exemplo, no Twitter, a localização geográfica do local de postagem das mensagens passou a ser disponibilizada apenas em 2010. Alguns trabalhos que abordam a descoberta de conhecimento espaço-temporal em dados do Twitter são mencionados a seguir.

Como os usuários usam bastante o Twitter para informar a seus seguidores o que estão fazendo no momento, esta rede tem características de tempo-real. Sakaki em [Sakaki et al 2010] usou esta característica para a detecção de eventos naturais como terremotos e tufões, usando as mensagens do Twitter como sensores, analisando as palavras das mensagens.

Um sistema para a descoberta de atividades sociais fora do padrão é proposto em [Lee et al 2011]. É utilizado o algoritmo K-means. Cada grupo formado é analisado considerando comportamentos de agregação (usuários que estavam em outros locais e agora estão neste) e dispersão (usuários que estavam neste local e agora estão em outros). Um pico nos dados de agregação é um indício de um evento social. Outro trabalho nesta área, mas que refina o processo com uma análise visual interativa foi proposto recentemente [Chae et al 2012]. O artigo [Lee 2012] vai mais além, pois preve a possível evolução e impacto dos eventos detectados.

3. A ferramenta Weka

Para a mineração e análise dos dados, utilizou-se o Weka [Witten & Frank 2005]. O Weka é uma ferramenta criada na Universidade de Waikato, Nova Zelândia, de código aberto, desenvolvido na linguagem de programação Java e muito utilizada nos meios acadêmicos.

Esta ferramenta possui uma coleção de algoritmos para execução das tarefas de mineração de dados.

A técnica utilizada neste trabalho foi a de Agrupamento (C*lustering*) e o algoritmo aplicado foi o DBSCAN [Ester et al 2006], que é um algoritmo baseado em densidade, isto é, as regiões densas formam os *clusters*. Para ser considerada densa, uma região deve ter um número mínimo de pontos (parâmetro “minPoints”) dentro de um círculo (parâmetro “epsilon”, raio do círculo).

Na ferramenta Weka, após a execução do algoritmo DBSCAN, é possível visualizar os resultados como mostra o exemplo da Figura 1.

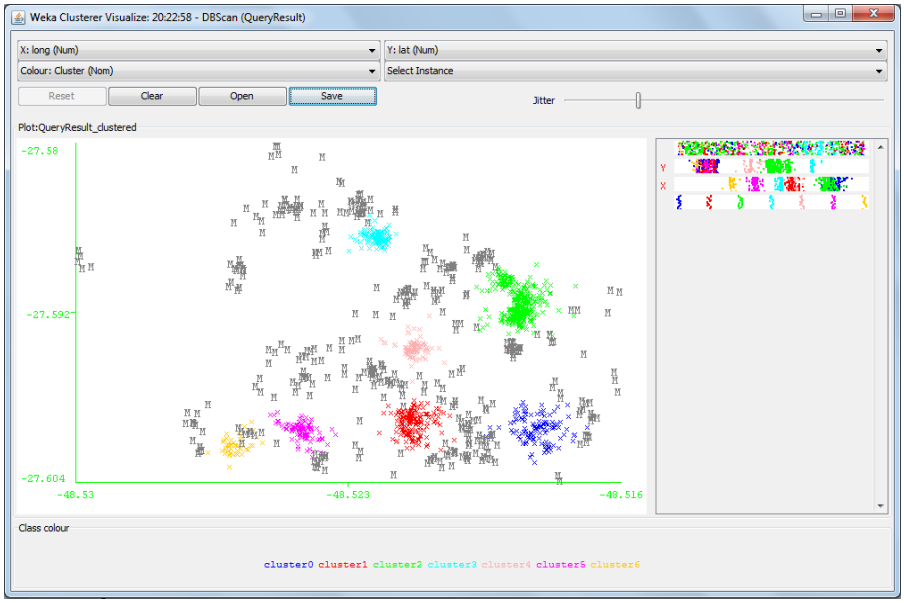


Figura 1. Visualização do DBSCAN na ferramenta Weka

Neste exemplo, é possível identificar os *clusters* que o algoritmo formou, neste caso 7. Os *clusters* são identificados pelas diferentes cores e, pode-se visualizar a posição de cada *cluster*, pois no eixo X foi plotado o atributo longitude e no eixo Y a latitude do ponto em que cada mensagem foi postada. Os pontos (*tweets*) que não pertencem a nenhum *cluster* aparecem na cor cinza e são considerados "ruído" ou *noise* pelo algoritmo.

O algoritmo DBSCAN, na ferramenta Weka, não possui recursos para trabalhar com dados geográficos. Desta maneira, é possível notar que seria difícil analisar este tipo de dado, pois não há informações geográficas, ou seja, não se tem como saber em que parte de uma cidade ou país está cada *cluster*. Para melhorar as análises, foram realizados melhoramentos no Weka, descritos na próxima seção.

4. Adaptações na Ferramenta Weka

Para que os resultados pudessem ser analisados de maneira mais ágil, sem que fosse necessário grande esforço humano, a ferramenta Weka foi adaptada. Para isto, foram desenvolvidos 2 recursos novos na ferramenta: (i) geração de um mapa onde cada marcador representa o centróide (centro de gravidade) de um *cluster*; (ii) com o clique de mouse em um marcador, podem ser visualizadas as palavras mais frequentes do *cluster* correspondente.

4.1. Geração de mapa com a API Google Maps

Para facilitar a análise de dados geográficos, optou-se por implementar a geração de um mapa real. Foi adicionado um método responsável por esta ação que é automaticamente executado durante a execução do algoritmo DBSCAN do Weka.

Para o desenvolvimento da geração do mapa, foi utilizada a API do Google Maps. Para a inserção de múltiplos pontos com ícones personalizados, foi utilizado como base o

script do site *Link Nacional* [Link Nacional 2011]. Com isso, foi possível indicar latitude, longitude, ícone/marcador e descrição para cada *cluster*.

No mapa desenvolvido, cada marcador representa o centróide de um *cluster*. Isto foi feito porque plotar todos os pontos de um *cluster* iria poluir muito o mapa e, com isso, dificultaria a análise. Além disso, o conjunto dos pontos de um *cluster* já pode ser visualizado na interface padrão do Weka, se houver necessidade de se conhecer melhor a distribuição dos pontos do *cluster*.

O resultado da geração do mapa é um arquivo HTML. A Figura 2 é um exemplo de como os *clusters* são visualizados na interface que foi desenvolvida neste trabalho. A interface mostra os centróides dos *clusters* gerados, representados pelos marcadores, que também identificam o período da mensagem (manhã, tarde ou noite) conforme a sua cor, e se foram postados em dias de semana ou nos fins de semana. No exemplo da Figura 2, todos os *clusters* são de mensagens postadas no período da manhã nos dias de semana.

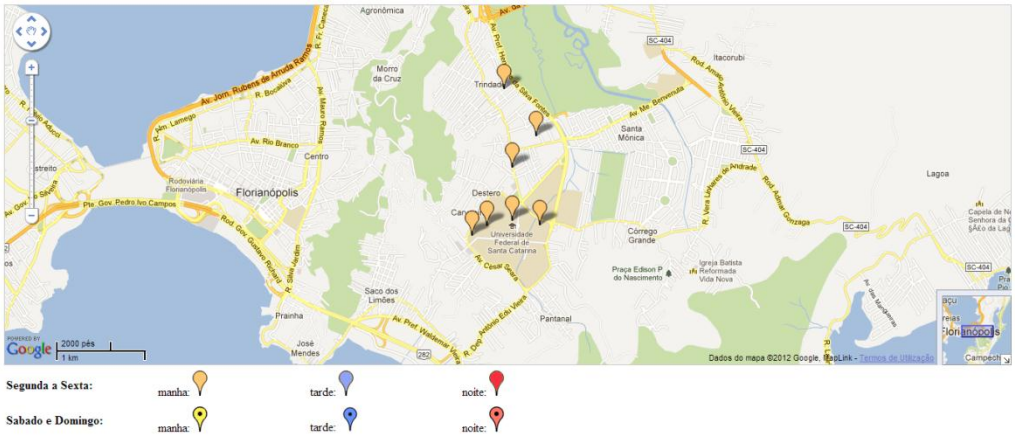


Figura 2. Visualização do mapa gerado

Conforme pode ser observado na Figura 2, o mapa desenvolvido facilita a análise de dados geográficos, pois é possível identificar onde cada *cluster* está situado no espaço, ou seja, é possível visualizar sobre qual local o *cluster* está localizado e também verificar nomes de ruas e bairros, a existência de rios, morros, etc.

No mapa gerado, os recursos do Google Maps podem ser utilizados (por exemplo, utilizar o *zoom*) e, além disso, foi implementada uma legenda com informações dos marcadores.

4.2. Obtenção de Palavras Frequentes

Para conhecer melhor o que os usuários do Twitter estão fazendo, foi implementada uma funcionalidade que captura as palavras mais frequentes das mensagens de cada agrupamento formado. Para isto, foi utilizada a biblioteca de tratamento de texto Tsearch2 [Bartunov & Sigaev 2012], que é uma extensão do PostgreSQL, desenvolvida na Universidade de Moscou. Optou-se por adaptar esta biblioteca em vez de desenvolver a funcionalidade, pois é uma tarefa complexa e que deve ser computacionalmente eficiente.

Antes de aplicar a função do Tsearch2, para a análise do texto do *tweet* em si, decidiu-se eliminar a acentuação ortográfica, para que a busca por texto encontrasse mais ocorrências de uma mesma palavra.

O Tsearch2 possui diversas opções para tratamento de texto, como eliminação de *stopwords*, *stemming*, etc. Para este estudo não foi realizado *stemming*, de modo que, por exemplo, os termos "casa" e "casarão" são considerados distintos. Foi utilizado o conjunto de *stopwords* (palavras ignoradas pelo sistema) referente à língua portuguesa, acrescentado de outras palavras observadas no decorrer do trabalho como irrelevantes para o estudo, como por exemplo, as letras isoladas e expressões como *4square*.

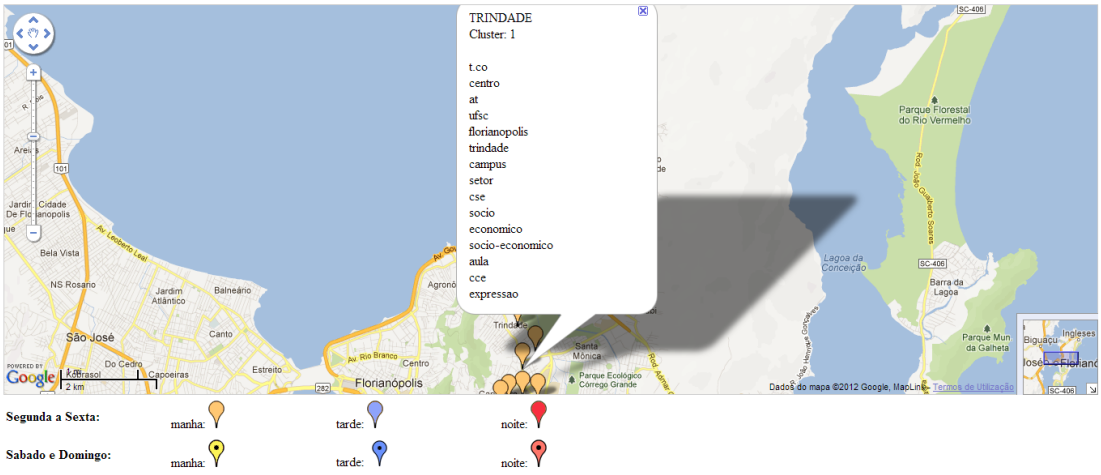


Figura 3. Visualização da interface com as palavras mais frequentes de um *cluster*

A Figura 3 apresenta a interface desenvolvida para a visualização das palavras mais frequentes nos *tweets* de um *cluster*. Basta clicar sobre um marcador para que a lista das palavras mais frequentes no cluster representado pelo marcador seja apresentada.

5. Experimentos Realizados

Para avaliar a eficácia das extensões realizadas foram realizados experimentos com *tweets* postados na cidade de Florianópolis, no período de abril a novembro de 2011, e contendo as coordenadas geográficas do local em que foram postados. Os *tweets* com as coordenadas geográficas corresponderam a aproximadamente 10% dos *tweets* emitidos. O SGBD utilizado foi o PostgreSQL. A escolha deste SGBD foi feita por este permitir a manipulação de dados geográficos por meio da extensão PostGIS, que segue o padrão OGC [OGC 2008].

As informações mais relevantes contidas na base de dados são: latitude e longitude (ambas do tipo "double"), data/hora de postagem da mensagem (tipo "timestamp") e texto do *tweet* propriamente dito (tipo "text").

Como havia a intensão de realizar a análise dos *tweets* por bairro de Florianópolis, uma primeira preparação dos dados foi a determinação do bairro em que os *tweets* foram postados. Para isto, inicialmente, foi criada na tabela de *tweets* uma coluna de tipo "geometry", necessária para a utilização das funções espaciais do PostGIS. Assim, para cada

tweet foi gerado um tipo geométrico “ponto”, por meio da função “ST_MakePoint” do PostGIS, aplicada aos campos latitude e longitude.

Em seguida, foi utilizado um arquivo *shapefile* disponível no *site* do Instituto Brasileiro de Geografia e Estatística (IBGE) para a obtenção dos limites dos bairros de Florianópolis. Desta maneira, foi empregada a função “ST_Contains” no PostGIS para o cruzamento da tabela de bairros e tabela de *tweets* para, enfim, popular a coluna com a informação do bairro em que o *tweet* foi postado.

Entre os 35 bairros de Florianópolis, 15 foram desconsiderados para a pesquisa em função do pequeno número de postagens realizadas. O total de registros (*tweets*) analisados foi de 152.552.

O conjunto de dados foi filtrado por 3 atributos: bairro, período do dia (manhã, tarde ou noite) e dia da semana (segunda a sexta-feira ou sábado e domingo), totalizando 6 consultas por bairro.

O algoritmo DBSCAN foi executado, inicialmente, de maneira padrão para os 20 bairros de estudo (minPoints = 2,5% dos *tweets* do bairro, epsilon = 0,045). Isto quer dizer que, para cada bairro, o algoritmo DBSCAN foi executado 6 vezes, totalizando 120 consultas no banco de dados aplicadas ao software Weka. Para que o estudo não se tornasse cansativo e para evitar o trabalho manual, o código da ferramenta Weka foi adaptado para automatizar estas execuções das consultas.

Como o algoritmo utilizado neste trabalho foi o DBSCAN, e o mesmo tem por característica gerar grupos em regiões densas, existe uma grande possibilidade de os pontos centrais gerados por *cluster* estejam sobre, ou muito próximos, a locais atrativos. Por exemplo, universidades, restaurantes, bares, shoppings centers, estádios de futebol, centros comerciais, empresas, entre outros. Isto pode ser percebido ao visualizar os marcadores plotados pelo Weka na interface desenvolvida neste trabalho.

Exemplos dos resultados obtidos com essa análise são mostrados nas Figuras 4 e 5. A Figura 4 apresenta os centróides de *clusters* no bairro Trindade. Pode-se observar que muitos *clusters* estão no campus da UFSC durante os dias de semana (marcado pelo círculo) e que nas noites de finais de semana muitos clusters são formados no entorno da UFSC (marcadores com ponto preto), que é uma região de muitos bares.

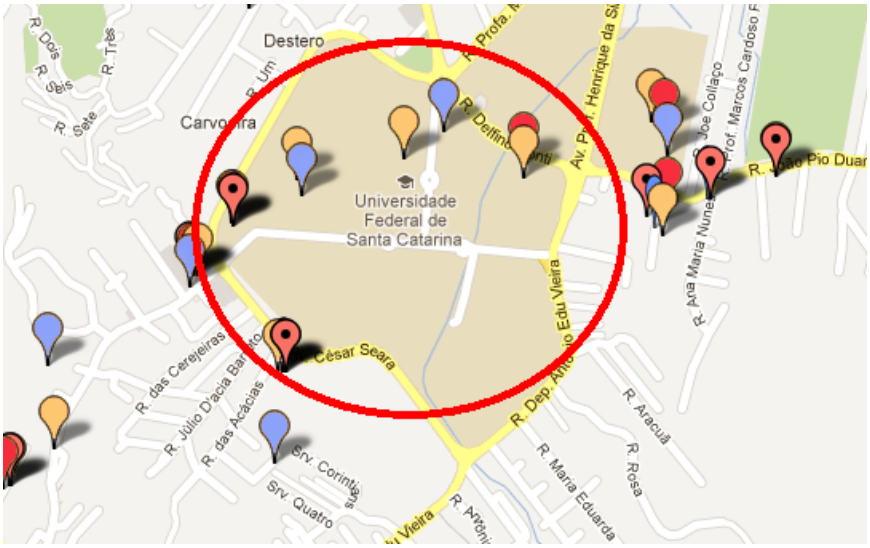


Figura 4. *Clusters* formados na UFSC e entorno

A Figura 5 apresenta *clusters* formados no estádio de futebol Orlando Scarpelli (marcado com o círculo) nas tardes e noites de finais de semana, o que deve corresponder a jogos sábados à noite e domingos à tarde. Além disso, esta figura apresenta as palavras mais frequentes encontradas em um dos *clusters*. Pode-se notar que estas palavras estão relacionadas a futebol.

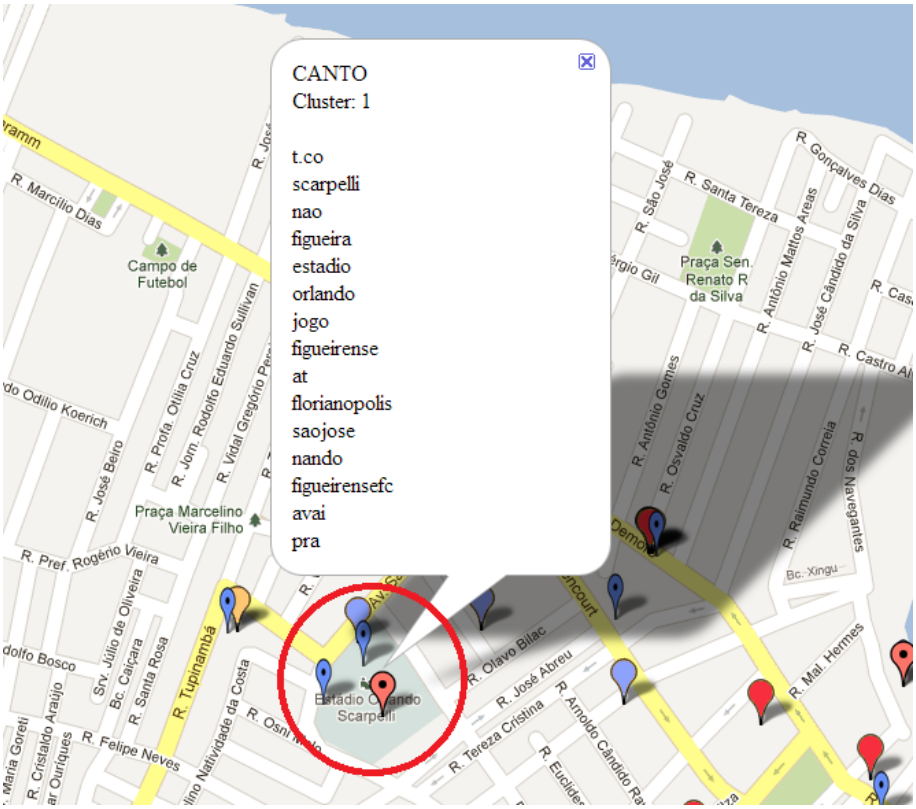


Figura 5. *Clusters* formados no estádio de futebol *Orlando Scarpelli*

Se para os demais bairros os parâmetros utilizados foram razoáveis, para o bairro Centro, a maioria das consultas gerou somente um *cluster* situado no meio deste bairro

(Figura 6). Os centróides ficaram aproximadamente no meio do bairro porque os dados eram muito numerosos e geograficamente homogêneos. A Figura 7 apresenta os *tweets* plotados no mapa, visualizado pela ferramenta Quantum GIS (<http://www.qgis.org>). Estes dados são somente do bairro Centro no período da tarde no intervalo de segunda a sexta-feira, totalizando 11.315 registros.

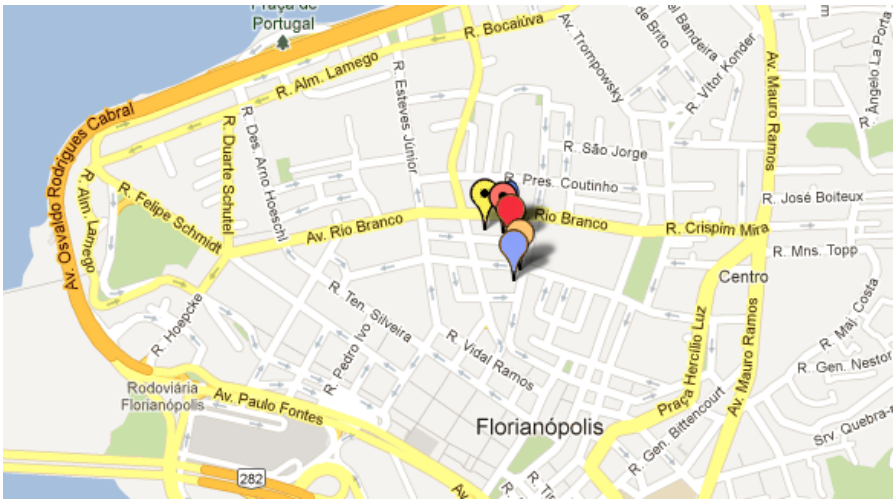


Figura 6. *Clusters* formados no bairro Centro

Para que o algoritmo DBSCAN possa gerar mais *clusters*, neste caso, é necessário diminuir o valor dos parâmetros "epsilon" e "minPoints". Por conseguinte, no bairro Centro, o algoritmo DBSCAN foi executado com diferentes valores de atributos, "épsilon" e "minPoints", até se tornar possível a descoberta de locais de interesse. Dois destes experimentos são detalhados na sequencia.

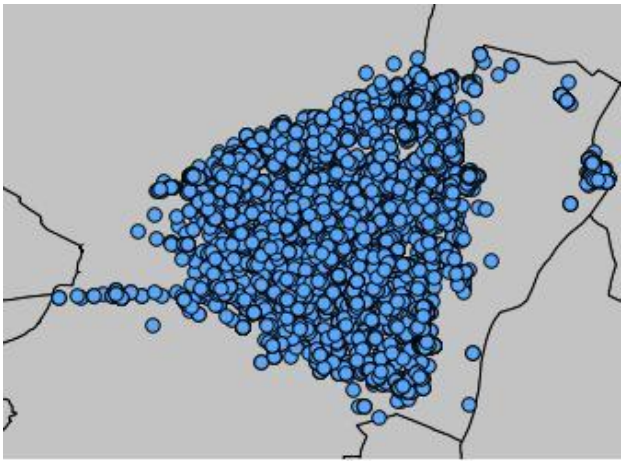


Figura 7. Visualização dos tweets do Centro através da ferramenta QuantumGIS

Para o experimento 1, foram utilizados os parâmetros: (i) minPoints = 1,25%; (ii) Épsilon = 0,011. Em relação às análises dos demais bairros, o número mínimo de pontos utilizado foi reduzido pela metade e o épsilon representou a quarta parte do valor utilizado nos experimentos com os outros bairros.

Com os parâmetros do experimento 1, foi possível identificar locais de interesse como: (i) Terminal de ônibus urbanos (TICEN) – períodos manhã e tarde nos dias de semana e noite tanto de dias de semana quanto de fins de semana; (ii) Instituto Estadual de Educação (IEE) – manhã e tarde de dias de semana; (iii) Praça XV de Novembro – manhã de fim de semana; (iv) Catedral Metropolitana de Florianópolis – tarde e noite de fim de semana; (v) Beiramar Shopping – manhã e tarde de dias de semana e fins de semana; (vi) Boate El Divino – tarde e noite de fins de semana; (vii) Boate 1007 – manhã e noite de fins de semana; (viii) Mercado Público – tarde de fins de semana; (ix) Morro da Cruz – manhã de fins de semana; (x) Instituto Federal de Santa Catarina (IF-SC) – tarde de dias de semana; (xi) Centro executivo localizado na Avenida Mauro Ramos – manhã de dias de semana, etc.

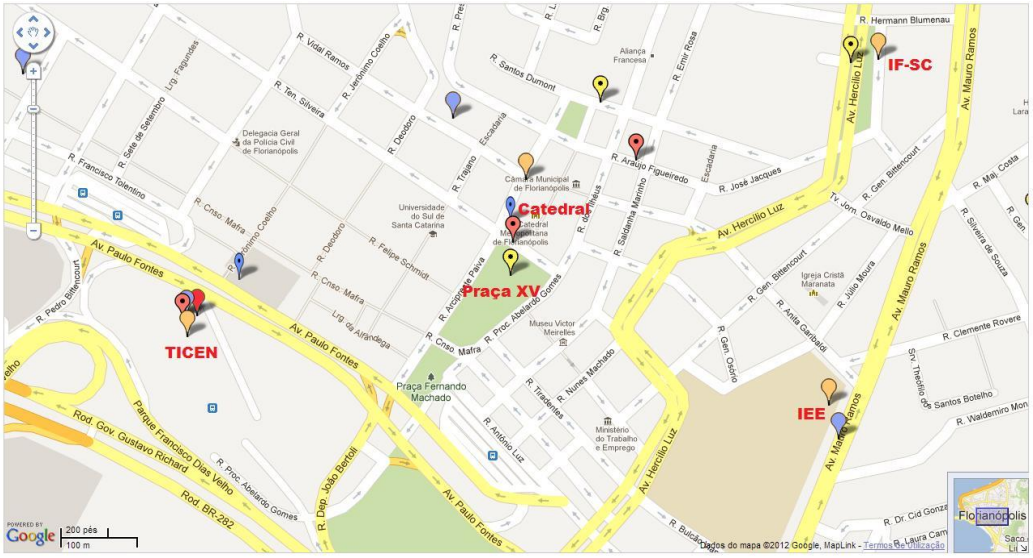


Figura 8. Resultado parcial da execução do experimento 1

A Figura 8 apresenta um zoom em parte do Centro com o resultado da execução do experimento 1. É possível identificar alguns dos locais citados, como o TICEN, a Catedral, o IF-SC, o IEE e a Praça XV.

Para o experimento 2, os parâmetros foram reduzidos radicalmente, com o objetivo de detectar um maior número de *clusters* que poderiam ser pequenos, em locais específicos. Foram utilizados os valores minPoints = 45 e epsilon = 0,0005.

Em relação à análise anterior (o experimento 1) foi observado, entre outros: (i) na Praça XV de Novembro (que foi considerada uma região densa no experimento anterior), por possuir uma área relativamente grande, não foi identificada como uma região densa, justamente por os *tweets* estarem mais distantes entre si neste local; (ii) alguns locais não detectados com análises anteriores puderam ser encontrados nesta análise, como a Universidade do Sul de Santa Catarina (UNISUL) e Terminal Rodoviário Rita Maria. Alguns clusters em locais residenciais também foram encontrados.

No mapa apresentado na Figura 9, gerado pela execução do experimento 2, é possível identificar alguns dos locais citados, como o Terminal Rita Maria e a UNISUL.

Figura 9. Resultado parcial da execução do experimento 2

6. Conclusão e Trabalhos Futuros

Mineração de dados espaço-temporais, com foco em detecção de agrupamentos, em redes sociais não é um assunto muito explorado. Esta pesquisa buscou conhecer o comportamento dos usuários do Twitter – especificamente na cidade de Florianópolis. De acordo com o conhecimento extraído, podem-se tirar conclusões do interesse da população e, analisar o que estão fazendo em determinados locais e horários. Por exemplo, donos de empresas, tendo acesso a estas informações, podem analisar a satisfação de colaboradores e/ou clientes. Este tipo de pesquisa poderá contribuir, por exemplo, com pesquisas de marketing, e consequentemente, aumentar a segurança dos resultados.

Para atender a proposta deste artigo, o código da ferramenta Weka foi adaptado. Isto tornou o *software* uma excelente ferramenta também para visualização. Desta maneira, o resultado encontrado pelo algoritmo DBSCAN pode ser melhor analisado.

Na implementação atual, pode ocorrer de um *cluster* ser formado apenas por *tweets* de um único usuário. Como trabalho futuro pode ser interessante tratar os dados para desconsiderar SPAMs, ou impedir a formação de um *cluster* se ele não contiver um número mínimo de usuários distintos.

Também se pretende permitir ações de usuário nos mapas gerados, como limpar *clusters* já populados no mapa e aplicar filtros para visualizar somente *clusters* de interesse. Permitir também mais flexibilidade ao usuário, adicionandos componentes na interface gráfica com este fim.

Referencias

Chae, J. Thom, D ; Bosch, H. ; Jang, Y. ; Maciejewski, R. ; Ebert, David S. ; Ertl, T. (2012) “Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition”. IEEE Conference on Visual Analytics Science and Technology (VAST). p 143-152.

Bartunov; Sigaev (2012). “Tsearch2 - full text extension for PostgreSQL”. http://www.sai.msu.su/~megera/postgres/gist/tsearch/V2/. Acessado em 30 de dezembro de 2012.

Ester, M.; Kriegel, H.-P.; Sander, J. and Xu, X. (1966) A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, Second International Conference on Knowledge Discovery and Data Mining, AAAI Press. p. 226-231.

Lee, C-H. (2012) Unsupervised and supervised learning to evaluate event relatedness based on content mining from social-media streams. Expert Syst. Appl. 39(18), p 13338-13356 .

Lee, C-H.; Yang, H.C.; Wen, W-S.; Weng, C-H. (2012) Learning to Explore Spatio-temporal Impacts for Event Evaluation on Social Media. ISNN (2), p 316-325.

Link Nacional. (2011). "Script de Múltiplos Pontos", http://www.linknacional.com.br/criar-site/2011/01/google-maps-api-multiplos-pontos-no-mapa-openinfowindowhtml. Acessado em 30 de dezembro de 2012.

OGC (2008) OpenGIS Standards and Specifications: Topic 5, Features. http://portal.opengeospatial.org/modules/admin/licenseagreement.php?suppressHeaders=0&accesslicense

Olhar Digital UOL. (2012) "Twitter gera meio bilhão de mensagens por dia", http://olhardigital.uol.com.br/jovem/redes_sociais/noticias/twitter-gera-meio-bilhao-de-tuites-por-dia. Acessado em 30 de dezembro de 2012.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes Twitter users: realtime event detection by social sensors. In Proceedings of the 19th international Conference on World Wide Web - WWW '10. ACM, New York, NY, p 851-860.

UOL Tecnologia. (2012) "Twitter passa dos 500 milhões de usuários, mas números mostram queda de microblog no Brasil", http://tecnologia.uol.com.br/noticias/redacao/2012/07/31/twitter-passa-dos-500-milhoes-de-usuarios-mas-numeros-mostram-queda-de-microblog-no-brasil.htm, Julho. Acessado em 30 de dezembro de 2012.

Witten, I. and Frank, E. (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco.