MAPEAMENTO DE MODELAGEM CONCEITUAL GEOGRÁFICA PARA ESQUEMA NOSQL HÍBRIDO

DANILO BOECHAT SEUFITELLI

MAPEAMENTO DE MODELAGEM CONCEITUAL GEOGRÁFICA PARA ESQUEMA NOSQL HÍBRIDO

Proposta de dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Mirella Moura Moro
Co-Orientador: Clodoveu Davis Jr.

Belo Horizonte

Junho de 2015

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Este documento apresenta o projeto de dissertação de mestrado em Ciência da Computação intitulado Mapeamento de Modelagem Conceitual Geográfica para Esquema NoSQL Híbrido. O objetivo principal da dissertação será projetar e analisar soluções em modelagem conceitual para dados geográficos, que utilize das abordagens não relacional e relacional, para construir uma modelagem híbrida. A Seção 1.1 descreve a motivação e justificativa para este trabalho; a Seção 1.2 mostra os objetivos gerais e específicos. O Capítulo 2 apresenta a revisão literária dos trabalhos relacionados. Por fim, o Capítulo 3 apresenta a metodologia do trabalho, bem como o cronograma de atividades.

1.1 Motivação e Justificativa

A cada dia que passa, mais pessoas se encontram conectadas à rede mundial de computadores e com os avanços tecnológicos há um considerável número de dispositivos tais como smartphones, tablets, notebooks, PDAs, entre outros, que são fontes incessantes de dados e informações, dentre elas, geográficas, devido ao acesso à internet bem como a integração do GPS por estes aparelhos.

Em meio a tantos dispositivos, uma nova categoria de aplicativos estão em desenvolvimento, tais como aplicações derivadas dos *Location-Based Services* (LBS) ou Serviços Baseados em Localização, que é um conceito recente que indica a integração de aplicações de localização geográfica (ou seja, coordenadas espaciais) com a noção geral de serviços [Schiller & Voisard, 2004].

Sistemas baseados em localização por sua vez, lidam com grande volume de dados e operações associadas a objetos geográficos, muitas vezes, mais complexas e custosas

que aplicações que não utilizam LBS. Como exemplo de aplicações fontes de dados geográficos temos o Facebook[1], Google Maps[2], Instagram[3], Waze[4], entre outros.

Tais sistemas necessitam de alta disponibilidade e recorrem a diversas soluções de gerenciamento de bancos de dados para atender a demanda crescente por geolocalização, baixo tempo de resposta e integração entre dispositivos. A maioria desses sistemas não utiliza os recursos tradicionais de bancos de dados relacionais, pois precisam de ganhos substanciais em desempenho e escalabilidade [Cho & Hwang, 2015].

Com o foco em desempenho e escalabididade para o armazenamento do grande volume de dados que as aplicações de grande porte demandam, surgiu o paradigma alternativo não relacional de banco de dados, popularmente chamado de NoSQL, que é o acrônimo para *Not Only SQL* ou Não Somente SQL. Esse paradigma se baseia na associação chave-valor, que pode variar a forma de armazenamento dos dados, partindo desde documentos (XML[5], RDF[6]), grafos, texto e notações orientadas a objetos, como por exemplo JSON[7] [Floratou et al., 2012].

Como forma de substituir o gerenciamento e armazenamento de dados de redes sociais, sistemas de informação e atualização em tempo real, que necessitavam de melhor desempenho, com acesso instantâneo a dados volumosos e heterogêneos, a tecnologia NoSQL surgiu a partir da Web 2.0, que descreve sites da *World Wide Web* que enfatizam o conteúdo gerado pelo usuário [Floratou et al., 2012].

A modelagem conceitual para dados geográficos, assim como para os tradicionais dados relacionais, é uma atividade importante pois permite a independência do esquema físico a ser utilizado bem como a reutilização da estrutura, ou parte dela, em consequência de que uma realidade geográfica modelada, pode se fazer presente em diferentes aplicações. Porém, há desafios nesta atividade devido às necessidades adicionais para a representação de dados geográficos, que são as propriedades geométricas e topológicas [Borges et al., 2005].

As propriedades geométricas são propriedades métricas, em que os relacionamentos métricos são definidos a partir de feições geométricas primitivas, tais como pontos, linhas e polígonos, os quais representam a geometria das entidades. Tais relacionamentos expressam a métrica das feições com referência a um sistema de coordenadas. De acordo com a geometria são estabelecidas algumas propriedades geométricas tais

[1]Facebook: http://www.facebook.com
[2]Google Maps: http://maps.google.com
[3]Instagram: https://instagram.com/
[4]Waze: https://www.waze.com/
[5]XML: eXtensible Markup Language
[6]RDF: Resource Description File
[7]JSON: JavaScript Object Notation http://www.json.org/

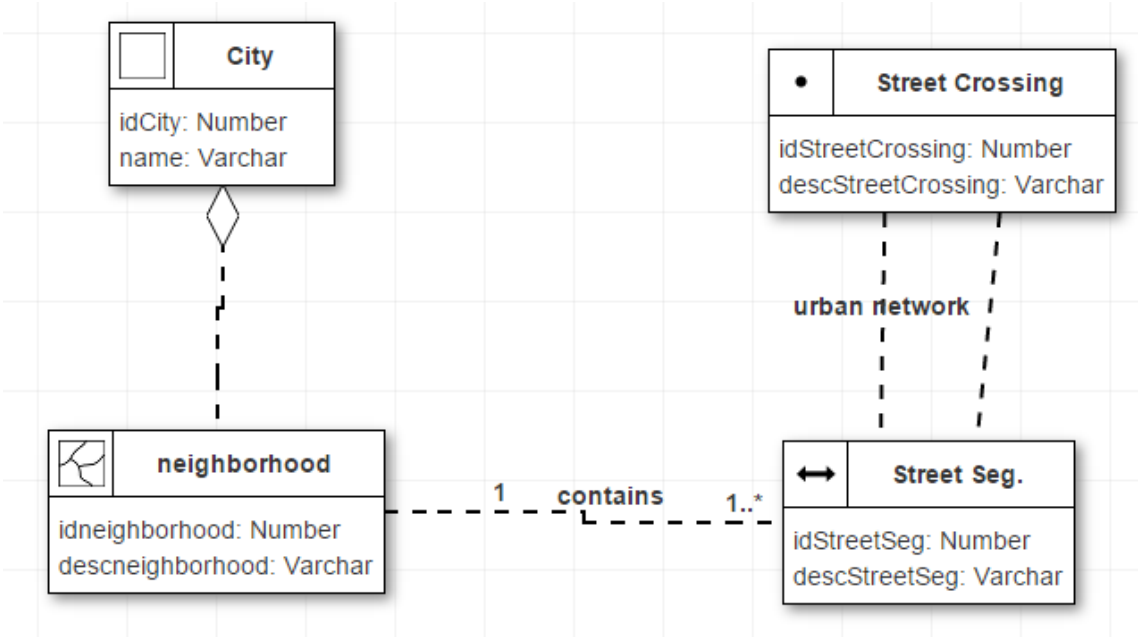

Figura 1.1. Esquema OMT-G: Representação de ruas e bairros

como, comprimento, sinuosidade e orientação para linha, perímetro e área da superfície para polígonos, volume para entidades tri-dimensionais, e forma e inclinação tanto para linhas quanto para polígonos [Laurini & Thompson, 1992].

Já as propriedades topológicas (não-métricas) são baseadas nas posições relativas dos objetos no espaço como conectividade, orientação (de, para), adjacência e contenção. Observa-se que alguns conceitos espaciais podem ser medidos tanto no domínio geométrico quanto no topológico. A proximidade, por exemplo, pode ser obtida tanto através de adjacência quanto da distância Euclidiana [Laurini & Thompson, 1992].

Uma solução SQL para modelagem conceitual de aplicações geográficas é proposta por Borges et al. [2001], com o OMT-G[8], que utiliza as primitivas definidas para o diagrama de classes UML (*Unified Modeling Language*) e introduz características geográficas com o objetivo de aumentar a capacidade de representação semântica daquele modelo. Deste modo, o OMT-G provê primitivas para modelar a geometria e a topologia dos dados geográficos, utilizando classes e relacionamentos espaciais [Borges et al., 2001]. A Figura 1.1 ilustra a modelagem conceitual geográfica para uma solução SQL, utilizando o OMT-G. Nela observa-se a representação dos bairros, cruzamentos e trechos de ruas de uma cidade através dos relacionamentos entre as entidades.

Esse esquema conceitual precisa ser mapeado para um esquema físico, como etapa

[8]OMT-G Designer: http://aqui.io/omtg/

```
"city":
{
  "idCidade": "ID_CIDADE",
  "nomeCidade": "CIDADE",
  "neighbohood":
  {
    "idNeighbohood": "ID_NEIGH",
    "descNeigh": "NEIGHBOHOOD",
    "StreetSeg":
    {
      "idStreetSeg": "ID_STREETSEG",
      "descStreetSeg": "DESC_STREETSEG",
      "StreetCross":
      {
        "idStreetCross": "ID_CROSS",
        "descCross": "DESC_CROSS"
      }

    }
  }
}
```

Figura 1.2. Esquema NoSQL *Document-Oriented* Hierárquico: Representação de ruas e bairros

indispensável da construção de um sistema de informação. Existem técnicas definidas para realizar esse mapeamento, tendo como destino um banco de dados relacional (ou objeto-relacional) geográfico, como o Oracle Spatial ou PostGIS. No entanto, não existem, até onde pôde-se determinar, propostas semelhantes para produzir esquemas físicos voltados para gerenciadores NoSQL.

Além disso, observa-se que os gerenciadores NoSQL têm características voltadas para determinadas estruturas de dados, como documentos (MongoDB[9]), hierarquias XML (eXist[10]) ou redes (Neo4J[11]), sendo que, hoje, nenhum deles combinam essas características com o suporte integral aos dados e funções necessárias para um BDG.

Porém, ao utilizar o paradigma NoSQL para a modelagem conceitual geográfica, não é possível obter um modelo conceitual único que satisfaça às necessidades das aplicações. Especificamente, NoSQL permite diferentes estratégias de armazenamento de dados, como por exemplo chave-valor, orientado a documentos, família de colunas e grafos [Sadalage & Fowler, 2012].

Dentre essas categorias não existe uma única forma de modelagem para o armazenamento de todas as representações dos objetos espaciais/geográficos. Por exemplo,

[9]MongoDB: https://www.mongodb.org/
[10]eXist: http://exist-db.org/exist/apps/homepage/index.html
[11]Neo4J: http://neo4j.com/

```
"city":
{
  "idCidade": "ID_CIDADE",
  "nomeCidade": "CIDADE",
  "descNeigh": "NEIGHBOHOOD",
  "descStreetSeg": "DESC_STREETSEG",
  "descCross": "DESC_CROSS"
}
```

Figura 1.3. Esquema NoSQL *Document-Oriented Flat* : Representação de ruas e bairros

considere o paradigma NoSQL orientado a documentos, em que os documentos são auto-descritivos e possuem uma estrutura hierárquica em árvore. As Figuras 1.2 e 1.3 apresentam diferentes formas de modelar a mesma realidade utilizando documentos com hierarquias diferentes (cada objeto geográfico em um subdocumento, ou todos no mesmo nível).

É importante destacar que cada modelagem tem seus problemas intrínsecos, por exemplo, na modelagem proposta representada pela Figura 1.2, há o possível problema de pesquisa em todos os subdocumentos, e já na modelagem apresentada pela Figura 1.3, há muita redundância dos dados, o que torna uma atualização destes dados uma atividade problemática, devido aos muitos descartes e tuplas novas.

Devido a essa diversidade de opções de modelagem, é necessário identificar qual categoria seria mais adequada para armazenar melhor cada objeto. Além disso, é necessário identificar o melhor mapeamento do modelo conceitual para o físico a fim de garantir o bom desempenho, eficiência e escalabilidade para aplicações geográficas. O que pode ser comprovado com os testes de desempenho realizado por nosso grupo de pesquisa[Onofre Santos, 2015], onde foram obtidos melhores resultados ao se utilizar modelagem em grafos com o gerenciador Neo4J, que possui o armazenamento em grafos de modo nativo, do que o PostgreSQL, que monta a estrutura do grafo no momento do SELECT.

1.2 Objetivos Gerais e Específicos

O objetivo geral deste projeto consiste em desenvolver uma metodologia de modelagem para dados geográficos, que seja capaz de combinar técnicas relacionais, como proposto por Borges et al. [2005], e técnicas não relacionais (NoSQL), de modo a identificar qual o tipo de modelagem atende melhor às necessidades particulares dos relacionamentos espaciais. Com isso, o intuito deste trabalho será a criação de uma metodologia com-

posta por sequências de atividades com o objetivo de modelar problemas reais que envolvam dados geográficos.

Pretende-se obter um esquema físico que combine os paradigmas relacional e não relacional, visto que tal escolha depende de vários fatores como, por exemplo, o tipo de dados que serão armazenados, o *workload*, a maior frequência de uso, por exemplo consulta ou atualização, dentre outros.

Para isso, os objetivos específicos são:

1. Identificar diversas técnicas de modelagem NoSQL para dados geográficos.
2. Avaliar qual das técnicas melhor funciona para cada tipo de relacionamento espacial.
3. Propor uma técnica de mapeamento que produza esquemas físicos híbridos combinando elementos relacionais e não relacionais.
4. Avaliar experimentalmente a nova metodologia híbrida proposta.

Capítulo 2

Trabalhos Relacionados

Este capítulo revisa a literatura relevante ao projeto, que inclui bancos de dados geográficos e bancos de dados NoSQL.

2.1 Banco de Dados Geográficos

Os anos noventa corroboraram com uma grande evolução na tecnologia de geoprocessamento, bem como um aumento significativo no número de Sistemas de Informação Geográfica (SIG) instalados. Tais sistemas foram aprovados, quer em ambientes de administração pública ou privada [Lisboa F. & Iochpe, 1999]. Nos dias de hoje, com o grande número de dispositivos capazes de registrar localizações geográficas, há o conceito de Serviços Baseados em Localização, do inglês, *Location-Based Services* (LBS), que são quaisquer serviços que levam em consideração a localização geográfica de uma entidade. O termo entidade significa que o objeto da informação sobre a localização pode ser humano ou não-humano. Por exemplo, um *pallet* de mantimentos é um objeto não-humano que muitas vezes precisa ser rastreado para fins de logística [Junglas & Watson, 2008]. Os LBSs podem ser definidos como serviços que integram a posição ou localização geográfica de um dispositivo móvel com outras informações para prover uma funcionalidade de maior valor agregado [Barkhuus & Dey, 2003].

Com estes sistemas, têm-se então os dados espaciais, que são quaisquer tipos de dados que descrevem fenômenos aos quais estejam associados a alguma dimensão espacial. Dados geográficos ou georreferenciados são dados espaciais em que a dimensão espacial está associada à sua localização na superfície da terra, num determinado instante ou período de tempo [Câmara, 1996].

Existem três características fundamentais para dados geográficos: características espaciais, não-espaciais e temporais. As características espaciais informam a posição

geográfica do fenômeno e sua geometria. As características não-espaciais descrevem o fenômeno e as características temporais informam o tempo de validade dos dados geográficos e suas variações sobre o tempo. A representação espacial de uma entidade geográfica é a descrição da sua forma geométrica associada à posição geográfica [Medeiros & Pires, 1994; Laurini & Thompson, 1992].

Os dados geográficos possuem propriedades geométricas e topológicas. As propriedades geométricas são propriedades métricas, em que os relacionamentos são definidos a partir de feições geométricas primitivas, tais como pontos, linhas e polígonos, os quais representam a geometria das entidades. Já as propriedades topológicas (não-métricas) são baseadas nas posições relativas dos objetos no espaço como conectividade, orientação (de, para), adjacência e contenção. Observa-se que alguns conceitos espaciais podem ser medidos tanto no domínio geométrico quanto no topológico [Laurini & Thompson, 1992].

Com isso, o modelo de dados deve ser o conjunto de conceitos que podem ser usados para descrever a estrutura e as operações em um banco de dados geográficos, onde o modelo busca por sistematizar o entendimento que é desenvolvido a respeito de objetos e fenômenos que serão representados em um sistema geográfico informatizado. Os objetos e fenômenos reais, no entanto, são complexos para permitir uma representação completa, considerando os recursos à disposição dos sistemas gerenciadores de bancos de dados (SGBD) atuais. Desta forma, é necessário construir uma abstração dos objetos e fenômenos do mundo real, de modo a obter uma forma de representação conveniente, embora simplificada, que seja adequada às finalidades das aplicações do banco de dados [Elmasri & Navathe, 1994].

O sucesso de qualquer implementação de um sistema de informação é sujeito da qualidade da transposição de entidades do mundo real e suas interações para um banco de dados informatizado, o que torna a abstração de conceitos e entidades existentes no mundo real, uma parte importante da criação de sistemas de informação. A abstração funciona como uma ferramenta que nos ajuda a compreender o sistema, dividindo-o em componentes separados, no qual cada componente pode ser visualizado em diferentes níveis de complexidade e detalhe, de acordo com a necessidade de compreensão e representação das diversas entidades de interesse do sistema de informação e suas interações.

Foi proposto por Borges et al. [2001], o OMT-G, um modelo de dados orientado a objetos para aplicações geográficas, que fornece primitivas para a modelagem da geometria e da topologia dos dados espaciais, ao qual suporta diferentes estruturas topológicas, múltiplas visões de objetos e também os relacionamentos espaciais. OMT-G também inclui ferramentas para especificar o processo de transformação e alternativas

de apresentação, que permitem, entre muitas possibilidades, modelar para múltiplas representações e apresentações.

Segundo Borges et al. [2001], o modelo OMT-G é baseado em três conceitos principais: classes, relacionamentos e restrições de integridade espaciais. Classes e relacionamentos definem as primitivas básicas usadas para criar esquemas estáticos de aplicação. OMT-G propõe o uso de três diferentes diagramas no processo de desenvolvimento de uma aplicação geográfica:

- **Diagrama de Classes:** todas as classes são especificadas junto com suas representações e relacionamentos. A partir deste diagrama, é possível derivar um conjunto de restrições de integridade espaciais, que deve ser observado na implementação.
- **Diagrama de Transformação:** todo o processo de transformação deve ser especificado, permitindo a identificação dos métodos necessários para a implementação, quando o diagrama de classes especifica múltiplas representações ou a derivação de uma classe a partir de outra.
- **Diagrama de Apresentação:** para especificar as alternativas de visualização que cada representação pode assumir.

Há também na literatura, o GeoFrame, que é uma alternativa proposta por Lisboa F. & Iochpe [1999] para a modelagem conceitual de dados geográficos. O GeoFrame é um framework conceitual que fornece um diagrama de classes básicas, que utiliza a notação gráfica da UML, para auxiliar o projetista na modelagem conceitual de fenômenos geográficos, bem como na especificação de padrões de análise para bancos de dados geográficos. O GeoFrame possui duas classes que são a base para qualquer aplicação geográfica, *THEME* e *GEOGRAPHICREGION*. Todos os aplicativos geográficos têm como principal objetivo a gestão e a manipulação de um conjunto de dados para uma determinada região de interesse, constituindo um banco de dados geográfico. Por exemplo, em um aplicativo GIS, a área urbana da cidade pode ser especificada como sendo a região geográfica de interesse (*GEOGRAPHICREGION*). Para esta região geográfica, pode-se imaginar que os seguintes temas (*THEME*) poderiam ser definidos: limites da zona urbana, rede rodoviária, bairros, edifícios (por exemplo, escolas, hospitais), transporte público, e zonas de coleta de lixo. É possível que alguns temas adicionais precisam ser definidos para algumas sub-áreas. Este seria o caso, por exemplo, de segurança especial e zoneamento de emergência para o centro da cidade.

Então, um esquema conceitual é construído a partir da especialização de classes do GeoFrame. A modelagem conceitual do banco de dados geográfico, usando Geo-

Frame, é realizada de acordo com uma abordagem *topdown* composto por três etapas. Inicialmente, para cada área geográfica considerada, vários temas (e subtemas) são identificados. Na segunda etapa, um diagrama de classe é especificado para cada tema identificado. A especificação das associações entre classes de diferentes temas ainda é feita nessa fase. Por fim, é feita a análise e modelagem do componente espacial de cada fenômeno geográfico [Lisboa F. & Iochpe, 1999].

Em suma, onde no passado tradicionalmente se usou SQL, hoje em diferentes cenários, a utilização destes sistemas pode deixar de prover a eficiência e eficácia necessárias para aplicações Web em larga escala, incluindo aquelas que utilizam dados geográficos e espaciais. E embora haja uma tendência para aumentar o compartilhamento de dados geoespaciais, principalmente com a ajuda de sistemas que funcionam através da Internet, pouco tem sido feito para facilitar a reutilização de soluções de modelagem de bancos de dados geográficos neste novo paradigma de banco de dados, o NoSQL.

2.2 Banco de Dados Não Relacionais (NoSQL)

Atualmente, o termo *Big Data* recebe atenção significativa em pesquisa e indústria. As características do *Big Data* geralmente são resumidas como os "4 Vs", que são *Volume, Velocidade, Variedade e Veracidade*. **Volume** descreve a enorme quantidade de dados armazenados, **velocidade** indica que a velocidade de mundança dos dados é muito alta, **variedade** denota que os dados podem ser armazenados em diversos formatos, por exemplo, estruturado e semi-estruturado e **veracidade** retrata a incerteza dos dados devido aos dados inconsistentes e incompletos [Hu & Dessloch, 2014b].

E com a nova geração de aplicações que surgiu para atender vários tipos de usuários, desde um pequeno grupo a grandes empresas, acabaram por produzir e manipular um volume massivo de dados, caracterizando assim o *Big Data* [Schreiner et al., 2015]. Com isso, a comunidade de banco de dados está atualmente em um ponto de inflexão sem precedentes. Por um lado, a necessidade de produtos de processamento de dados nunca foi mais elevada do que o nível atual, e por outro lado, o número de novas soluções de gestão de dados que estão disponíveis explodiu nos últimos anos. Durante a última década, como a indústria em quase todos os setores da economia mudou-se para um mundo orientado a dados, tem havido uma explosão no volume de dados, e a necessidade de ferramentas de processamento de dados mais ricos e mais flexíveis [Floratou et al., 2012].

Nesse cenário, os bancos de dados não relacionais (NoSQL) foram projetados

para o crescimento horizontal escalável, de modo a prover uma grande quantidade de operações de leitura e escrita por segundo. Em oposição aos SGBDs tradicionais que não escalam tão bem quando distribuídos em diversos servidores. Segundo Cattell [2011] as principais características desses sistemas são:

- Escalar horizontalmente adicionando novos servidores;
- Ser replicável e distribuído entre vários servidores;
- Interface simples ou protocolo de acesso (diferente de complexas linguagens SQL);
- Um sistema de paralelismo e concorrência mais fraco que as transações nos bancos relacionais (opcionalidade das travas de leitura, escrita e compartilhadas);
- Distribuição eficiente de índices e uso de memória;
- A possibilidade de se alterar atributos de registros dinamicamente.

Bancos de dados NoSQL possuem quatro arquiteturas de armazenamento de dados para lidar com diferentes tipos de dados e, portanto, são adequados para diferentes casos de utilização [Sadalage & Fowler, 2012; Kaur & Rani, 2013]. São elas:

- Chave-Valor;
- Orientado a Colunas;
- Orientado a Documentos;
- Orientado a Grafos.

A arquitetura de armazenamento por **chave-valor** utiliza o modelo de índices por chave-valor. Esse é o tipo de banco de dados NoSQL mais simples. O conceito dele é uma chave e um valor para essa chave. Este método persiste os dados em um modo não estruturado (*schema-less*). Assim, os dados podem ser armazenados em um tipo de dados de uma linguagem de programação ou um objeto [Saxena et al., 2014]. A Figura 2.1 apresenta a migração do esquema conceitual relacional da Figura 1.1 para um esquema NoSQL chave-valor. Os bancos de dados mais populares que suportam este modelo de armazenamento são o Riak[1], Redis[2], Memcached DB[3], HamsterDB[4], Amazon DynamoDB[5] e o Projeto Voldemort[6] [Sadalage & Fowler, 2012].

[1]Riak: http://basho.com/riak/
[2]Redis: http://redis.io/
[3]Memcached DB: http://memcached.org/
[4]HamsterDB: http://hamsterdb.com/
[5]Amazon DynamoDB: http://aws.amazon.com/pt/dynamodb/
[6]Projeto Voldemort: http://www.project-voldemort.com/voldemort/

```
{
    idCity: "ID_CITY",
    city: "CITY",
    neighborhood:
    [
        {
            idNeighborhood: "ID_NEIGHBORHOOD",
            descNeighborhood: "NEIGHBORHOOD",
            street:
            [
                {
                    idStreetSeg: "ID_STREET",
                    descStreetSeg: "DESC_STREET",
                    cross:
                    [
                        {
                            idStreetCross: "ID_CROSS",
                            descCross: "DESC_CROSS"
                        }
                    ]
                }
            ]
        }
    ]
}
```

Figura 2.1. Esquema *Key-Value*: Representação de ruas e bairros

Um banco de dados NoSQL **orientado a colunas** organiza os dados de uma maneira estruturada e armazena os dados que pertencem à mesma coluna de forma contínua no disco, ao contrário de bancos de dados relacionais, onde as linhas que são armazenadas de forma contígua. Tais sistemas são projetados para atender três seguintes aspectos: grande número de colunas, natureza escassa de dados e mudanças frequentes no esquema. A mudança no projeto de armazenamento resulta em um melhor desempenho para algumas operações como agregações, o suporte para *ad-hoc*, consulta dinâmica e etc. Em bancos de dados *row-oriented*, todas as colunas dessas linhas que satisfaçam a cláusula WHERE da consulta são recuperados, o que provoca desnecessários acessos ao disco, caso apenas algumas colunas forem exigidas dentre todas as colunas retornadas. A maior parte desses bancos de dados colunares também são compatíveis com a estrutura *MapReduce*, o que acelera o processamento de uma grande quantidade de dados, de modo a distribuir o problema em grande número de sistemas [Saxena et al., 2014; Kaur & Rani, 2013; Hu & Dessloch, 2014a].

A Figura 2.2 apresenta a migração do esquema conceitual relacional da Figura 1.1 para um esquema NoSQL orientado a colunas. Os bancos de dados orientado a

```
//super column family
{
    //row (key)
    idCity: "ID_CIDADE",
    value:
    {
        city: "CIDADE",
        idNeighborhood: "ID_NEIGHBORHOOD",
        value:
        {
            neighborhood: "NEIGHBORHOOD",
            IDstreetSeg: "id_STREET_SEG",
            value:
            {
                streetSeg: "STREET_SEG",
                idStreetCross: "ID_STREET_CROSS",
                value:
                {
                    streetCross: "STREET_CROSS"
                }
            }
        }
    }
}
```

Figura 2.2. Esquema *Column-Oriented*: Representação de ruas e bairros

coluna de código aberto populares são MonetDb[7], Hypertable[8], HBase[9] e Cassandra[10]. Hypertable e HBase são derivados de BigTable onde, como Cassandra tem suas características de ambos BigTable e Dynamo [Kaur & Rani, 2013].

O termo documento de um banco de dados **orientado a documentos** refere-se ao conjunto solto de pares de chave/valor, geralmente JSON, em vez dos tradicionais documentos e tabelas. Estes documentos são auto descritivos, com uma estrutura hierárquica em árvore, que pode conter mapas, coleções e valores escalares. O banco de dados orientado a documentos se refere ao documento como um todo, em vez de dividir o documento em muitos pares de chave/valor. Isso permite que os documentos de diferentes estruturas sejam colocados em um mesmo conjunto. O banco de dados orientado a documentos suporta índice de documentos, incluindo não apenas identificadores primários, mas também propriedades do documento [He, 2015].

A Figura 2.3 apresenta a migração do esquema conceitual relacional da Figura

[7]MonetDB: https://www.monetdb.org/Home
[8]Hypertable: http://hypertable.com/home/
[9]HBase: http://hbase.apache.org/
[10]Cassandra: http://cassandra.apache.org/

```
"city"
{
    idCity: "ID_CITY",
    nameCity: "CITY",
    neighborhood:
    {
        idNeighborhood: "ID_NEIGHB",
        descNeighborhood: "NEIGHBORHOOD",
        streetSeg:
        {
            idStreetSeg: "ID_STREETSEG",
            descStreetSeg: "DESC_STREETSEG",
            streetCross:
            {
                idStreetCross: "ID_CROSS",
                descCross: "DESC_CROSS"
            }
        }
    }
}
```

Figura 2.3. Esquema *Document-Oriented*: Representação de ruas e bairros

1.1 para um esquema NoSQL orientado a documentos. Os exemplos típicos de bancos de dados orientados a documentos incluem MongoDB[11], CouchDB[12] e Terrastore[13].

Por fim, outra categoria *schema-less* para o armazenamento de dados não relacionais, são os chamados bancos de dados **orientado a grafos**, em que basicamente consiste na coleção de nós e arestas. Cada nó representa uma entidade (tal como uma pessoa ou empresa) e cada aresta representa uma ligação ou relação entre dois nós. Cada nó em um banco de dados em grafo é definida por um elemento identificador único, um conjunto de arestas de saída e/ou bordas de entrada e de um conjunto de propriedades expressos como pares de valores-chave. Cada aresta é definida por um elemento identificador único, um nó de partida e/ou um nó de destino e um conjunto de propriedades. Bancos de dados em grafo aplicam a teoria dos grafos para o armazenamento de informações sobre as relações entre entradas. As relações entre pessoas em redes sociais, é o exemplo mais óbvio. As relações entre os itens e atributos em motores de recomendação, é outra. Bancos de dados relacionais não são capazes de armazenar dados de relacionamentos entre pessoas, e esses tipos de consultas podem

[11]MongoDB https://www.mongodb.org/
[12]CouchDB: http://couchdb.apache.org/
[13]Terrastore: https://code.google.com/p/terrastore/

```
Node city = graphDB.creatNode();
city.setProperty("name", "CITY");

Node neighborhood = graphDB.creatNode();
neighborhood.setProperty("name", "NEIGHBORHOOD");

city.createRelationshipTo(neighborhood, "neighborhood");

Node streetCross = graphDB.Node();
streetCross.setProperty("name", "STREET_CROSS");

Node streetSeg = graphDB.Node();
streetSeg.setProperty("name", "STREE_SEG");

streetCross.createRelationshipTo(streetSeg, "urbanNetwork");
streetSeg.createRelationshipTo(streetCross, "urbanNetwork");

neighborhood.createRelationshipTo(streetSeg, "street");
streetSeg.createRelationshipTo(neighborhood, "street");
```

Figura 2.4. Esquema *Graph-Oriented*: Representação de ruas e bairros

ser complexas, lentas e imprevisíveis. Os bancos de dados em grafos são projetados para este tipo de problema, o que torna as consultas mais confiáveis [Hashem & Ranc, 2015].

A Figura 2.4 apresenta a migração do esquema conceitual relacional da Figura 1.1 para um esquema NoSQL orientado a grafos para o gerenciador Neo4J. Os principais bancos de dados orientados a grafos são o Neo4j[14], HypergraphDB[15] e AllegroGraph[16].

Com base em NOAM (NoSQL Abstract Model), um modelo de dados abstrato inovador para bancos de dados NoSQL que explora as semelhanças de vários sistemas NoSQL, também utilizado para especificar uma representação independente do sistema de dados do aplicativo, Bugiotti et al. [2014] propuseram uma metodologia de design de banco de dados para sistemas NoSQL. Especificamente, esta metodologia tem o objetivo de projetar uma "boa"representação de dados de aplicativos em um banco de dados NoSQL, de modo a apoiar a escalabilidade, desempenho e consistência, como necessário por aplicações web da nova geração. Os experimentos mostraram que o design de bancos de dados NoSQL deve ser feito com cuidado, pois afeta consideravelmente o desempenho e a coerência das operações de acesso aos dados e sua metodologia fornece uma ferramenta eficaz para a escolha entre diferentes alternativas.

Devido às características de dados espaciais, como explicado na seção anterior, juntamente com a flexibilidade dos bancos de dados não relacionais, não há nenhuma técnica de modelagem de dados geográficos que sirva para todas as aplicações, como um

[14]Neo4j: http://neo4j.com/
[15]HypergraphDB: http://www.hypergraphdb.org/index
[16]AllegroGraph: http://allegrograph.com/

modelo único e ideal, pois a implementação de tecnologias NoSQL são muito diversas.

Soluções NoSQL podem prover a eficiência necessária para aplicações que utilizam dados geográficos. Amirian et al. [2013] fornecem um levantamento das características dos grandes volumes de dados geoespaciais e possíveis soluções para a sua gestão e tratamento. Eles apontam uma visão geral dos principais tipos de soluções NoSQL, suas vantagens e desvantagens e os desafios que elas apresentam na gestão desses grandes volumes de dados, na elaboração de um servidor de dados geoespaciais usando o padrão de serviços web geoespaciais com um banco de dados XML NoSQL como um *backend*.

Outro importante trabalho relacionado à modelagem de dados NoSQL, devido às características da natureza dinâmica dos dados clínicos, que são frequentemente organizados hierarquicamente e armazenados como texto e número livre, Lee et al. [2013] analisaram três modelos de banco de dados: NoSQL, XML e XML nativo para avaliar a adequação dos dados clínicos estruturados pelos seguintes aspectos: o desempenho da consulta, escalabilidade e extensibilidade. Os resultados mostraram que banco de dados NoSQL é a melhor escolha para a velocidade da consulta, enquanto bancos de dados XML são vantajosos em termos de escalabilidade, flexibilidade e extensibilidade, que são essenciais para lidar com as características dos dados clínicos, que são dinâmicos, esporádicos e heterogêneos em essência.

Para auxiliar na decisão de qual porção dos dados de uma empresa é persistido como XML e que parte como dados relacionais, Moro et al. [2007] descrevem o ReXSA, uma ferramenta que aborda o desafio de projetar esquemas de banco de dados híbridos. Dado uma anotação do modelo de informação dos dados de uma empresa, o ReXSA avalia e recomenda um esquema de banco de dados que harmoniza modelos de dados relacionais e XML, um problema até então não abordado pela literatura e que representa uma séria necessidade na indústria. Tem a vantagem de considerar propriedades qualitativas do modelo de informação como a reutilização, evolução e perfis de desempenho para decidir como persistir os dados.

Com isso, tem-se que a modelagem dos dados para o armazenamento em banco de dados não relacionais, seja para dados geográficos ou não, é uma atividade não trivial, pois cada aplicação exige uma modelagem que se enquadre às suas necessidades, não havendo um modelo único que satisfaça todas as exigências de diferentes aplicações de um mesmo seguimento. Entretanto, uma boa modelagem promove características como escalabilidade, flexibilidade e extensibilidade e proporciona também a reutilização do esquema, ou boa parte dele.

Capítulo 3

Metodologia

Este capítulo apresenta a metodologia do projeto. Serão apresentados os métodos previstos, plano de atividades e cronograma.

3.1 Métodos Previstos

A proposta desta dissertação consiste na criação e avaliação de diferentes modelagens de dados geográficos para base de dados não relacionais (NoSQL) para as quatro arquiteturas de armazenamento NoSQL: Chave-Valor, Orientado a Colunas, Orientado a Documentos e Orientado a Grafos.

As modelagens não relacionais propostas terão como ponto de partida a modelagem SQL provenientes da modelagem em OMT-G [Borges et al., 2005]. Devido a isso será necessário o desenvolvimento de um conversor de dados que mantenha as características das entidades, relacionamentos e objetos modelados em OMT-G do banco original, sem alterar seu conteúdo, independentemente do formato. O conversor deverá ser capaz de traduzir dados relacionais contidos em tabelas para não relacionais contidos nas quatro arquiteturas de armazenamento NoSQL.

A avaliação das modelagens não relacionais propostas serão avaliadas em dois momentos. Primeiramente será necessária a criação de métricas de avaliação, para identificar os casos em que cada tipo de modelagem proposta pelo projeto se sobressai. E em segundo momento, com as melhores abordagens de modelagem não relacional identificadas, pretende-se comparar com as abordagens encontradas na literatura, que servirão como *baseline* para os experimentos.

Por fim, pretende-se propor uma modelagem híbrida para dados geográficos, em que se concilie dados armazenados em SQL e NoSQL, de modo a obter o máximo de desempenho para aplicações geográficas.

3.2 Plano de Atividades

As fases para realização desta proprosta foram divididas em etapas, as quais são listadas a seguir.

1. Revisão bibliográfica
    a) Continuação dos estudos das arquiteturas NoSQL.
    b) Continuação dos estudos de modelagens NoSQL.
    c) Continuação dos estudos sobre métricas de avaliação de desempenho.
2. Execução da metodologia
    a) Definição das modelagens NoSQL possíveis.
    b) Avaliação das modelagens identificadas.
    c) Proposição da modelagem híbrida.
3. Formulação da métrica proposta
    a) Definição da métrica.
    b) Implementação.
    c) Experimentação e análise.
4. Etapas permanentes
    a) Atualização da revisão literária.
    b) Escrita da dissertação.
    c) Defesa da dissertação.

3.3 Cronograma

O cronograma apresentado na Tabela 3.1 foi elaborado a partir do plano de atividades apresentado na seção 3.2. As etapas foram distribuídas ao longo do tempo estimado para a realização de cada tarefa.

Tabela 3.1. Cronograma de atividades proposto.

Etapa	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Revisão Bibliográfica															
Estudos arquiteturas NoSQL	x	x													
Estudos modelagens NoSQL	x	x													
Estudo métricas avaliação	x	x													
Execução da Metodologia															
Definição das modelagens NoSQL		x	x	x											
Avaliação das modelagens			x	x	x	x									
Proposição da modelagem híbrida			x	x	x	x	x	x	x						
Formulação da Métrica Proposta															
Definição da métrica							x	x	x						
Implementação							x	x	x	x					
Experimentação e análise									x	x	x	x	x	x	
Etapas Permanentes															
Atualização da revisão literária	x	x	x	x	x	x	x	x	x	x	x	x	x		
Escrita da dissertação	x	x	x	x	x	x	x	x	x	x	x	x	x		
Defesa da dissertação															x

Referências Bibliográficas

Amirian, P.; Basiri, A. & Winstanley, A. (2013). Efficient online sharing of geospatial big data using nosql xml databases. Em *Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on*, pp. 152--152. IEEE.

Barkhuus, L. & Dey, A. K. (2003). Location-based services for mobile telephony: a study of users' privacy concerns. Em *INTERACT*, volume 3, pp. 702--712. Citeseer.

Borges, K. A. V.; Davis, C. A. & Laender, A. H. (2001). Omt-g: an object-oriented data model for geographic applications. *GeoInformatica*, 5(3):221--260.

Borges, K. A. V.; Davis Jr., C. A. & Laender, A. H. F. (2005). Modelagem conceitual de dados geográficos. Em Casanova, M.; Câmara, G.; Davis Jr., C. A.; Vinhas, L. & Ribeiro, G., editores, *Banco de Dados Geográficos*, pp. 83–136. MundoGeo Editora.

Bugiotti, F.; Cabibbo, L.; Atzeni, P. & Torlone, R. (2014). Database design for nosql systems. Em Yu, E.; Dobbie, G.; Jarke, M. & Purao, S., editores, *Conceptual Modeling*, volume 8824 of *Lecture Notes in Computer Science*, pp. 223--231. Springer.

Cattell, R. (2011). Scalable sql and nosql data stores. *SIGMOD Rec.*, 39(4):12--27.

Cho, Y. C. & Hwang, S. (2015). Future trends in spatial information management: Suggestion to new generation (internet of free-open). *International Journal of Signal Processing Systems*, 3:75–81.

Câmara, G. (1996). Caracterização de dados geográficos. Em Câmara, G.; Casanova, M. A.; Hemerly, A. S.; Magalhães, G. C. & Medeiros, C. M. B., editores, *Anatomia de Sistemas de Informação Geográfica*, pp. 37–48. Instituto de Computação - UNICAMP.

Elmasri, R. & Navathe, S. (1994). *Fundamental of Database Systems*. Addison-Wesley, Menlo Park, CA, 2nd edição.

Floratou, A.; Teletia, N.; DeWitt, D. J.; Patel, J. M. & Zhang, D. (2012). Can the elephants handle the nosql onslaught? *Proceedings of the VLDB Endowment*, 5(12):1712--1723.

Hashem, H. & Ranc, D. (2015). An integrative modeling of bigdata processing. *International Journal of Computer Science and Applications*, 12(1):1–15.

He, C. (2015). Survey on nosql database technology. *Journal of Applied Science and Engineering Innovation Vol*, 2(2).

Hu, Y. & Dessloch, S. (2014a). Defining temporal operators for column oriented nosql databases. Em Manolopoulos, Y.; Trajcevski, G. & Kon-Popovska, M., editores, *Advances in Databases and Information Systems*, volume 8716 of *Lecture Notes in Computer Science*, pp. 39–55. Springer International Publishing.

Hu, Y. & Dessloch, S. (2014b). Extracting deltas from column oriented nosql databases for different incremental applications and diverse data targets. *Data & Knowledge Engineering*, 93(0):42 – 59. Selected Papers from the 17th East-European Conference on Advances in Databases and Information Systems.

Junglas, I. A. & Watson, R. T. (2008). Location-based services. *Commun. ACM*, 51(3):65--69.

Kaur, K. & Rani, R. (2013). Modeling and querying data in nosql databases. Em *Big Data, 2013 IEEE International Conference on*, pp. 1--7. IEEE.

Laurini, R. & Thompson, D. (1992). *Fundamentals of Spatial Information Systems*, volume 37. London: Academic Press.

Lee, K. K.-Y.; Tang, W.-C. & Choi, K.-S. (2013). Alternatives to relational database: comparison of nosql and xml approaches for clinical data storage. *Computer methods and programs in biomedicine*, 110(1):99--109.

Lisboa F., J. & Iochpe, C. (1999). Specifying analysis patterns for geographic databases on the basis of a conceptual framework. Em *Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems*, GIS '99, pp. 7--13, New York, NY, USA. ACM.

Medeiros, C. B. & Pires, F. (1994). Databases for gis. *ACM Sigmod Record*, 23(1):107--115.

Moro, M. M.; Lim, L. & Chang, Y.-C. (2007). Schema advisor for hybrid relational-xml dbms. Em *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 959--970. ACM.

Onofre Santos, P. (2015). On the efficiency of relational, document and graph data models for managing mobile spatial data. Dissertação de mestrado, Universidade Federal de Minas Gerais, Belo Horizonte - MG.

Sadalage, P. J. & Fowler, M. (2012). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.

Saxena, M.; Ali, Z. & Singh, V. K. (2014). Nosql databases-analysis, techniques, and classification. *Journal of Advanced Database Management & Systems*, 1(2):13--24.

Schiller, J. & Voisard, A. (2004). *Location-Based Services*. Elsevier.

Schreiner, G. A.; Duarte, D. & dos Santos Mello, R. (2015). Análise de abordagens para interoperabilidade entre bancos de dados relacionais e bancos de dados nosql. Em *XI Escola Regional de Banco de Dados - ERBD*, Caxias do Sul, RS.