

PONTIFÍCA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**Um estudo sobre a modelagem conceitual de bancos de
dados NoSQL**

Gustavo Carvalho da Silva Pampillón Rodrigues

PROJETO FINAL DE GRADUAÇÃO

**CENTRO TÉCNICO CIENTÍFICO – CTC
DEPARTAMENTO DE INFORMÁTICA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

Rio de janeiro, novembro de 2023



Gustavo Carvalho da Silva Pampillón Rodrigues

**Um estudo sobre a modelagem conceitual de bancos de
dados NoSQL**

Relatório de Projeto Final 2, apresentado ao programa
Ciência da Computação da PUC-Rio como requisito
parcial para a obtenção do título de Bacharel em Ciência
da Computação.

Orientador: Marcos Vianna Villas

Rio de Janeiro,
novembro de
2023

RESUMO

Rodrigues, Gustavo. Villas, Marcos. Um estudo sobre a modelagem de banco de dados NoSQL. Rio de Janeiro, 2023. 30 páginas. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro

Bancos de dados NoSQL são uma categoria especial de banco de dados que difere em vários aspectos do banco de dados relacional. O objetivo desse trabalho tem como foco a etapa da modelagem conceitual de tais bancos de dados, de forma a entender o que vem sendo feito nessa situação e ao final sugerir um modelo conceitual que possa auxiliar a modelagem lógica de tais bancos de dados, chave-valor, documento, grafos e relacional. A partir do estudo de diversos artigos científicos e material elaborados por acadêmicos da área foi feita uma análise extensa que a partir dela foi feita a proposta de modelo. Ao final com algumas extensões ao modelo ER, foi criada uma camada intermediária entre a modelagem conceitual e lógica que incorpora diversos modelos de dados em uma única representação.

Palavras-Chave: banco de dados NoSQL; modelo conceitual; ER.

ABSTRACT

Rodrigues, Gustavo. Villas, Marcos. A study about the conceptual Modeling of NoSQL databases. Rio de Janeiro, 2023. 30 pages. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro

NoSQL databases are a special category of databases which differs in many aspects of the relational database. The goal of this project focuses on the conceptual modeling of such databases, in a way to understand what has been done to solve this situation and at the end suggest a conceptual model which can assist the logical modeling of these databases, key-value, documents, graphs and relational. From the study of several scientific articles and material prepared by academics in the field an extensive analysis was made, from which was made a model proposal. At the end with some extensions to the ER model, it was created an intermediate layer between the conceptual and logic modeling steps that incorporates several data models in a single representation.

Keywords: NoSQL databases; conceptual model; ER.

Sumário

1. Introdução	5
2. Situação Atual	7
3. Objetivo.....	8
4. Metodologia	8
5. Estudo Inicial.....	9
5.1 Bancos de Dados Chave-Valor	9
5.2 Bancos de Dados de Grafos	10
5.3 Banco de Dados Orientado a Documentos	12
6. Revisão Bibliográfica	14
7. Proposta Inicial de Modelo Conceitual Poliglota	18
7.1 Do ER para Documento.....	19
7.3 ER -> Chave-Valor.....	21
8. Análise de Extensões.....	22
8.1 Extensões para Grafos.....	22
8.2 Extensões para Documentos	23
8.3 Extensões para Chave-Valor	23
9. Proposta de Modelo Conceitual Poliglota.....	24
9.1 Foco em Grafo	24
9.2 Foco em Documentos	25
9.3 Foco em Chave Valor.....	26
9.4 Multi Modelo.....	26
10. Validação do Modelo Proposto	28
11. Exemplo de Aplicação	30
12. Conclusão.....	31
13. Referências Bibliográficas	32

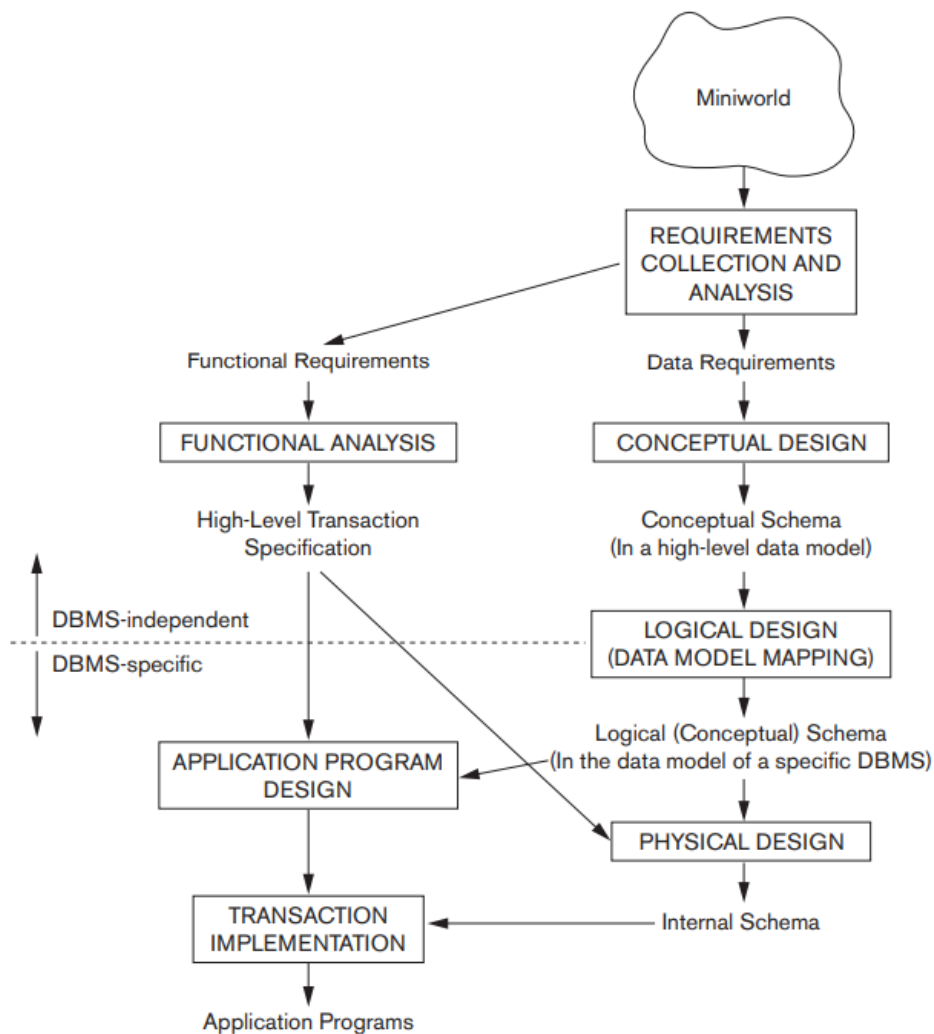
1. Introdução

Em seu livro sobre fundamentos de bancos de dados, Elmasri e Navathe [1] elencam as etapas de processo de projeto (design) de um banco de dados, onde são descritas 3 etapas de modelagem (Figura 1).

A primeira etapa consiste num modelo conceitual que descreve de forma concisa os requisitos dos dados e descreve tipos de entidades, relações e restrições de integridade. Tal primeira etapa consiste num esquema, uma representação simplificada e funcional que representa o banco, que deve ser preparado em alto nível e pode ser feito usando uma variedade de técnicas. O curso de Ciência da Computação da PUC-Rio apresenta duas possíveis técnicas para a modelagem conceitual: o Modelo de Entidades e Relacionamentos, proposto por Chen [17], e o Diagrama de Classes da UML (Unified Modeling Language) [18].

Na segunda etapa, de modelagem lógica, são utilizados modelos de dados tradicionais como o modelo relacional, descrito por Codd [2], e também existem modelos mais antigos e rudimentares, como o modelo utilizados em base de dados em rede e hierárquicas [23]. Já os bancos de dados OO se aproveitam da UML, modelo criado para conversar com programação OO, o banco de dados em rede pode ser modelado com um simples grafo e o banco hierárquico é naturalmente descrito em forma de uma árvore.

Figura 1



Fonte: ELMASRI, Ramez e NAVATHE, Shamkant B. Sistemas de Banco de Dados. Pearson Addison Wesley. 7a Edição, 2019.

Em 1998 foi criado o termo NoSQL para se referir à novos bancos de dados que usavam diferentes abordagens de armazenamento para atender aplicações com contextos mais específicos ou contornar limitações do tradicional banco relacional. Nos dias de hoje os bancos NoSQL já figuram $\frac{1}{3}$ dos bancos mais usados no mercado [3], porém todos ainda lidam com o mesmo problema: a falta de consenso quando se trata de modelagem. Muitos dos bancos recentes NoSQL não possuem um modelo que consegue descrevê-los nessa etapa conceitual, o que gera um grande problema logo em uma das primeiras etapas do projeto de um banco de dados.

2. Situação Atual

Bancos de dados NoSQL vem se tornando cada vez mais populares por atender demandas não supridas pelos SGBDs relacionais, assim como servem para aplicações de contexto mais específico, ou inclusive o oposto, aplicações mais simples que não demandam um banco tão robusto e poderoso.

Entre os bancos NoSQL mais usados está o orientado a documentos que se tornou extremamente popular por trabalhar com formatos de dados (descrição e conteúdo) muito usadas, sendo elas XML (*Extensible Markup Language*) [24] e JSON (*Javascript Notation Object*) [25]. Ambas tiveram surgimento anos depois do banco relacional, logo os bancos orientados a documentos vieram para suprir a necessidade do armazenamento de dados nesse formato [26] que vem crescendo bastante desde que surgiram.

Outro banco de dados NoSQL que vem ganhando espaço nos últimos anos é o banco de dados de grafos, que se encaixa naqueles bancos que surgiram para aplicações de contexto mais específico. Tais bancos têm sido amplamente usados, entre outros contextos, para detecção de fraude e modelagem de grandes redes, em especial redes sociais, e sistemas de recomendação.

Tais bancos compartilham um mesmo problema: a falta de uma convenção de modelo conceitual. Isso quer dizer que atualmente qualquer projeto de banco de dados NoSQL ou não possui modelo conceitual ou é gerado uma forma particular de se modelar o banco em questão. Uma solução comum que tem sido utilizada é a criação de MER estendidos, que são criados exclusivamente para cada projeto de banco de dados feito, o que dificulta o entendimento dessas modelagens, pois toda vez que um desenvolvedor diferente quer modelar seu banco ele cria sua própria forma de modelagem conceitual ou escolhe um entre dezenas de modelos propostos que mais atende às suas necessidades.

Atualmente já podem ser identificadas diversas tentativas de criação própria de uma modelagem conceitual, feitas em diversas partes do mundo, como por exemplo, o modelo proposto por Imam, Basri, Ahmad e Gonzáles-Aparicio [4], que propõe um modelo para bancos de dados orientados a documentos, assim como o DOT, uma linguagem descritiva de grafos desenvolvida pela empresa americana AT&T [27]

Outra questão importante de ser abordada é a necessidade de uma modelagem conceitual que possa ser traduzida para diferentes modelos lógicos, isto é, um mesmo modelo que possua diferentes seções dele mesmo que representem diferentes modelos lógicos, como relacional, documento, grafo entre outros. Essa necessidade surge de grandes sistemas que usam mais de um tipo de banco de dados diferente, mas precisam

ser representados em um único modelo. Como exemplo principal exemplo dessa necessidade temos as plataformas de e-commerce, que lidam com dados de diferentes formas e estruturas em múltiplos contextos [41].

3. Objetivo

O objetivo do presente trabalho foi avaliar como realizar a modelagem conceitual de bancos de dados tendo em vista a sua posterior modelagem lógica para modelos NoSQL, chave-valor, documentos e grafos, seja esta modelagem lógica homogênea (apenas um modelo lógico) ou não (multi-modelo). Em relação a modelagem lógica homogênea, este trabalho tem como primeiro objetivo descobrir uma forma de como modelar os bancos de dados alvo do trabalho, chave-valor, documento e grafo, a homogeneidade se refere ao tipo de modelo lógico em questão, onde num primeiro momento apenas um tipo desses alvos será abordado por vez. O segundo e principal objetivo foi uma abordagem multi-modelo, onde um mesmo modelo conceitual pode representar até quatro modelos distintos, chave-valor, documento, grafos e o tradicional relacional.

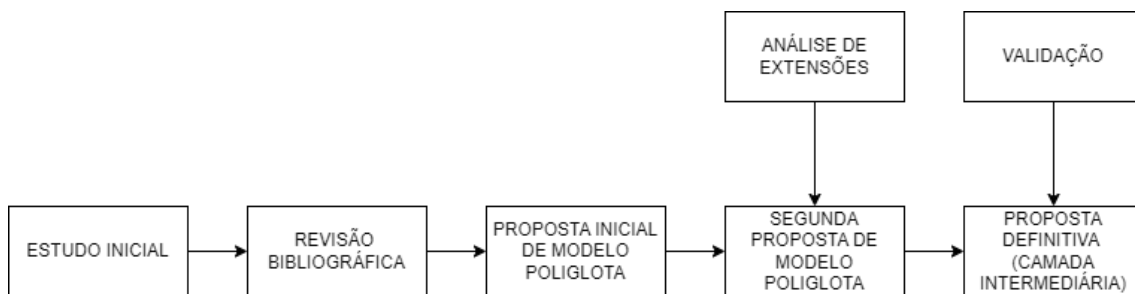
4. Metodologia

Para o desenvolvimento deste trabalho a metodologia aplicada se deu em diversas etapas. Primeiro foi feito um estudo inicial das tecnologias envolvidas, assim como um levantamento bibliográfico de artigos que discutem a modelagem dos bancos estudados, seja usando modelos já estabelecidos como ER e UML e possíveis extensões, até os que implementam um modelo completamente novo. A implementação de um modelo próprio foi baseada em duas etapas, a primeira foi um primeiro olhar na possível tradução de um modelo ER para cada tipo de banco, sem nenhuma extensão e também sem olhar nenhuma referência externa, trabalhando do zero.

Em seguida quando esses rascunhos de modelos já foram desenhados foi feita uma análise dos artigos levantados a fim de aplicar soluções já encontradas por outros trabalhos nesses primeiros modelos a fim de aprimorá-los e resolver questões que até então não foram consideradas. Feitos os aprimoramentos no modelo foi estudado a possibilidade de um só modelo que pode ser traduzido para os três tipos de bancos no escopo deste trabalho. Ao fim foi proposta um exemplo de aplicação a fim de ilustrar a

efetividade do modelo criado. Na figura 2 temos um fluxograma que ilustra a metodologia utilizada.

Figura 2



5. Estudo Inicial

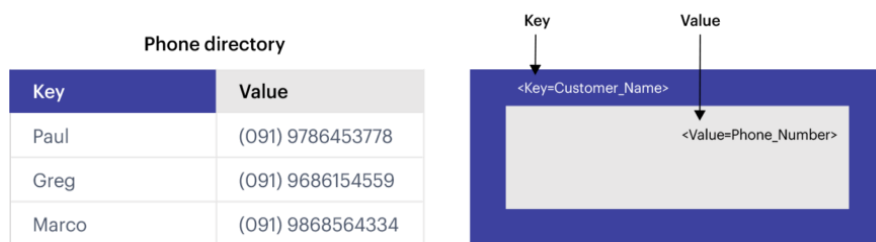
O estudo inicial teve como objetivo alcançar um melhor entendimento dos principais objetos de estudo deste trabalho, isto é, bancos de dados NoSQL de três tipos: chave-valor, bancos de dados em grafo e bancos de dados orientado a documentos. A pesquisa a respeito desses bancos foi focada em entender seus históricos, motivações em relação a sua criação, os objetos com o qual esses bancos trabalham, como tais objetos são manipulados, assim como exemplos dos SGBDs mais populares de cada tipo de banco. Nessa etapa foi decidido que não seria avaliado bancos de dados do tipo colunar, pois esse tipo de banco é uma outra forma de implementação a partir do modelo relacional, logo consideramos que não faz sentido agrupá-lo junto com os bancos NoSQL em análise.

5.1 Bancos de Dados Chave-Valor

Os bancos de dados chave-valor são bancos projetados para armazenar estruturas de dados simples, geralmente algo nos moldes uma tabela hash, que consiste numa simples associação entre uma chave identificadora e o dado em si. Algo a se notar em bancos de dados chave-valor é que não existe uma linguagem de busca e não é possível fazer uma busca que considere o dado armazenado, pois é possível apenas busca dados considerando os valores das chaves. As únicas operações possíveis para alteração de dados são adição e remoção de pares chave-valor. Essa limitação vem em troca de uma alta performance onde o acesso ao dado é essencialmente imediato.

Bancos de dados chave-valor funcionam bem com aplicações que lidam com um alto fluxo de operações simples de leitura e escrita, ou seja, operações, sem a necessidade de consulta, devido ao seu rápido acesso à memória. Um dos principais exemplos de aplicação recomendada é gerenciamento de sessões de interação em sistemas, como por exemplo em aplicações web, em larga escala, que por natureza tem alta taxa de transferência de dados e caráter *schemaless*. O principal banco de dados chave-valor atual é o Redis [3], que armazena todos os dados em memória principal, amplificando todas as vantagens anteriormente citadas que esse tipo de banco possui. Além do Redis algumas das maiores empresas de tecnologia possuem seus próprios SGBDs chave valor sendo alguns deles: DynamoDB da Amazon, o RocksDB do Facebook e o LevelDB da Google. Na figura 3 há um pequeno exemplo que apresenta dados no formato chave-valor.

Figura 3



Fonte: <https://redis.com/wp-content/uploads/2020/10/key-value-figure-b-v2-1024x314.png?&auto=webp&quality=85,75&width=1200>

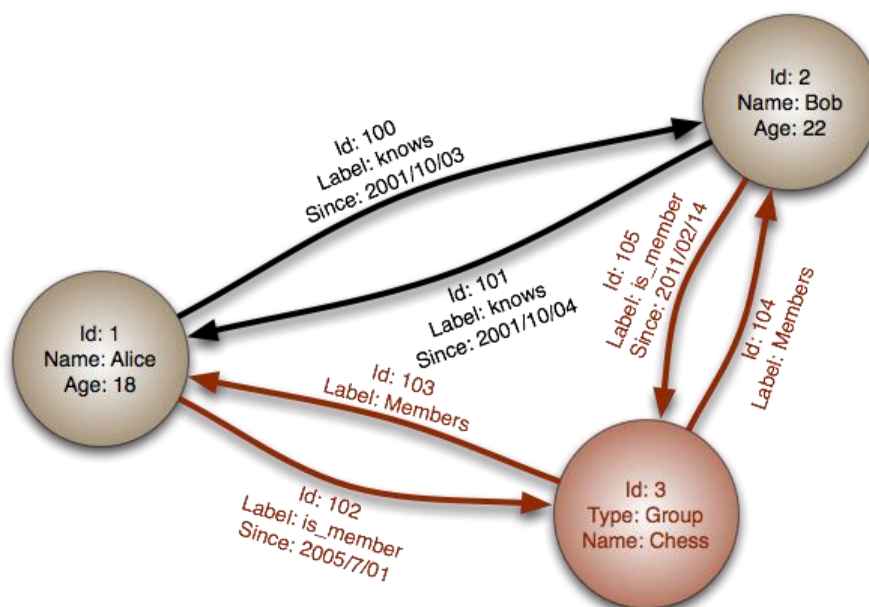
5.2 Bancos de Dados de Grafos

No fim da década de 60, fazendo uso dos bancos de dados em rede, era possível criar estruturas com as mesmas características de grafos, mas com tempo surgiu a necessidade de bancos específicos para esse tipo de estrutura de dado. Tal banco é formado por três objetos principais: nó, aresta e atributos. O nó é a representação do dado em si, podendo por sua vez ter atributos que o caracteriza, já as arestas são as ligações entre nós e possuem um nó inicial e final, além de direção e também um ou mais atributos que a caracteriza. A combinação desses objetos gera uma estrutura navegável, que é o grande apelo do banco de dados em grafo. Essas características geram condições favoráveis para aplicações bem específicas, e entre os exemplos mais citados estão detecção de fraude, modelagem de redes sociais e aplicações no campo da biomedicina [28, 29]. Existem 2 principais tipos de grafos, grafos rotulados com

propriedades, onde os nós e arestas recebem nomes ou rótulos arbitrários e os grafos do tipo RDF (Resource Description Framework) [21], onde os nós recebem um formato de tripla, e apresenta uma estrutura de sujeito, objeto e predicado, feitos por um conjunto de nó aresta e nó.

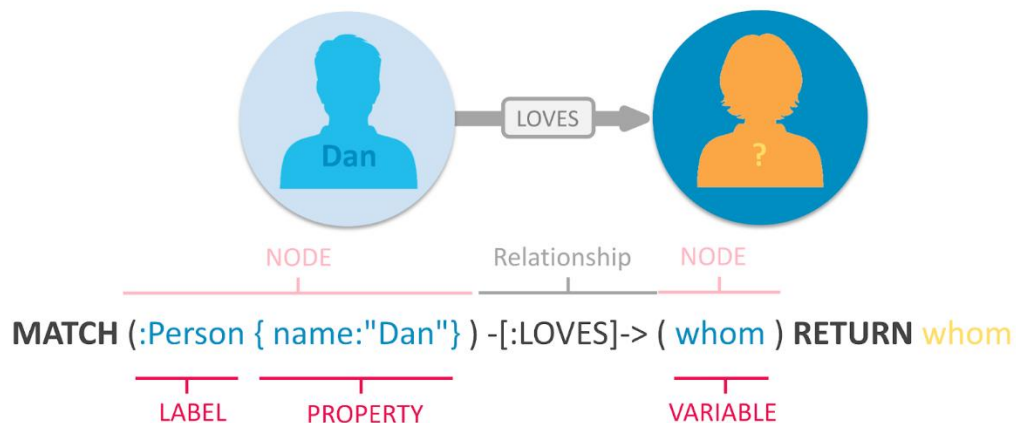
Apesar dos bancos em grafos permitirem buscas, não existe uma linguagem unificada como o sql, fazendo com que cada SGBD diferente tenha sua própria forma de consulta, como por exemplo, o Cypher do Neo4J e o Gremlin, desenvolvida pelo Apache. O Neo4J, um dos bancos de grafos mais populares também implementa o conceito de *labels*, rótulos em tradução livres, que são um quarto atributo que é usado para caracterizar nós com base num domínio ou tipo que pode ser estabelecido. Na figura 4 podemos um pequeno exemplo de um grafo com 3 nós, 6 arestas e seus respectivos atributos. Também podemos observar que o nó de id 1 possui três propriedades e não está rotulado, enquanto a aresta de id 100 possui 2 propriedades e 1 label, “knows”. Em sequência temos a figura 5 que exemplifica uma consulta em Cypher, linguagem de consulta utilizada pelo Neo4J e mencionada acima o comando busca todas as conexões do nó “Dan” que são feitas pela aresta do tipo “loves”, via uma variável chamada “whom”.

Figura 4



Fonte: https://upload.wikimedia.org/wikipedia/commons/3/3a/GraphDatabase_PropertyGraph.png

Figura 5



Fonte: <https://dist.neo4j.com/wp-content/uploads/sample-cypher.png>

5.3 Banco de Dados Orientado a Documentos

Bancos de dados orientados a documentos estão entre os mais populares dos bancos NoSQL e foram criados para armazenar e lidar com documentos. Documentos nesse contexto são uma forma de apresentar dados semiestruturados, ou seja, uma combinação de dados com e sem estrutura que possui características dos dois [26], onde um conjunto de dados é formado por um par de rótulos e dados. Entre os formatos de documentos mais populares estão XML e JSON. O conceito de documento também é acompanhado do conceito de coleções, que são um conjunto de documentos. Em algumas aplicações ainda é possível implementar documentos que pertencem a mais de uma coleção.

Bancos orientados a documentos não possuem uma linguagem universal de busca, dependendo então de APIs que são disponibilizadas por cada SGBD diferente, fator esse que dificulta a portabilidade entre bancos e o quanto um novo desenvolvedor precisa aprender para poder trabalhar com essa ferramenta. Por outro lado, esse aumento do nível de aprendizado diminui quando se trata do dado em si, visto que documentos são um formato de dado muito popular [24, 25] e a sua característica de ser semiestruturada provê uma grande flexibilidade na criação de novas entradas no banco assim como promove uma boa sinergia com aplicações externas devido a popularidade do formato. O principal SGBD de banco de documentos é o MongoDB [3], porém existem outras distribuições mantidas e desenvolvidas por grandes empresas, como o DocumentDB da Amazon, Firestore da Google, assim como distribuições open source, como é o caso do Couchbase.

Abaixo temos na figura 6 um exemplo de um documento, onde podemos observar a presença de atributos que possuem estrutura, como o campo “address” e campos desestruturados, que é o caso do campo “hobbies” e também podemos observar o campo “_id” que é uma identificação daquele documento gerada de forma automática pelo MongoDB, e logo na sequência na figura 7 um diagrama que demonstra a arquitetura de um banco orientado a documentos, onde no topo temos onde o dado é armazenado, seguido do conceito de coleções, que por sua vez vem seguida de um conjunto de documentos. Na figura 8 temos um exemplo de busca no MongoDB, onde os atributos “name” e “address” são buscados numa coleção específica, com o critério de idade superior a 18 e também um limite de 5 no resultado.

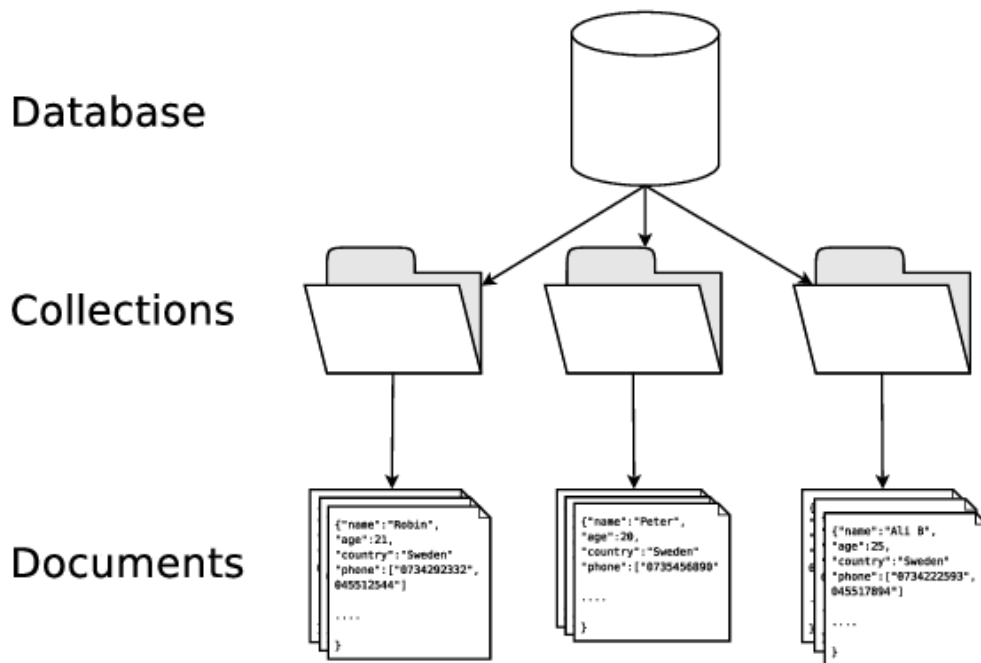
Figura 6



```
{
  "_id": "5cf0029cafff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

Fonte: https://webimages.mongodb.com/_com_assets/cms/kq6uev0p1jnpq1ob3-snippet_light.svg?auto=format%2Ccompress&ch=DPR

Figura 7



Fonte: HENRICSSON, Robin. Document Oriented NoSQL Databases: A comparison of performance in MongoDB and CouchDB using a Python interface. 2011

Figura 8

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
)<div data-bbox="605 554 838 629" data-label="Text">

← collection  
← query criteria  
← projection  
← cursor modifier


```

Fonte: <https://www.mongodb.com/docs/manual/images/crud-annotated-mongodb-find.bakedsvg.svg>

6. Revisão Bibliográfica

O objetivo dessa revisão era, entre outros, responder uma pergunta: O que está sendo feito para contornar a falta de modelo conceitual em bancos NoSQL? Primeiramente foi utilizada uma estratégia de busca mais direcionada, procurando artigos e estudos que lidam com os tipos de bancos que são alvos deste trabalho o que gerou um grande número de artigos para serem lidos e analisados sem ao menos saber se o artigo de fato responderia à pergunta proposta. Eventualmente foi encontrado um artigo [15] que analisou outros 54 artigos publicados entre 2008 e 2019, e apesar de ter

um escopo maior do que foi proposto como objetivo do presente relatório ele foi muito importante para responder à pergunta aqui proposta.

O artigo de referência para a revisão bibliográfica [15] realizou um mapeamento sistemático da modelagem de dados em bancos NoSQL, isso quer dizer que além de analisar a modelagem conceitual, também é feita uma análise das etapas de modelagem lógica e física, o que não nos interessa, pois, essas etapas de modelagem fogem do escopo do trabalho. Isso quer dizer que nem todo artigo coletado fala sobre a etapa de modelagem conceitual, diminuindo a quantidade de objetos de estudo, mesmo assim servindo como um grande facilitador. A seguir é apresentada uma visão geral das estratégias usadas para modelar bancos NoSQL, conforme consta no artigo de referência [15]:

Orientado a documento:

- Nova notação, baseada em ER
- Nova notação, baseada em UML
- UML
- OWL
- FCA

Grafo:

- ER Estendido
- UML
- Modelo Genérico
- RDF
- O-ER
- Nova notação, baseada em ER
- ER
- OWL

Chave-Valor:

- Nova notação, baseada em ER
- OWL
- UML

Colunar:

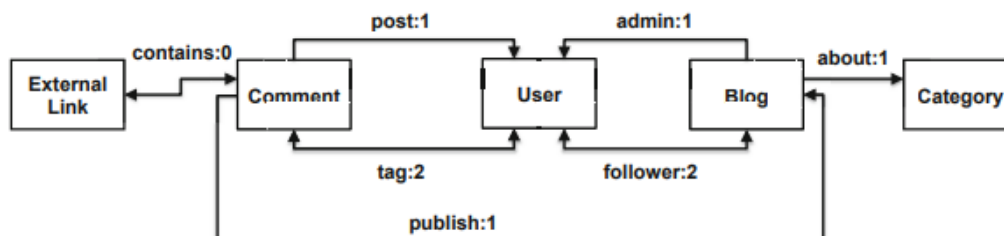
- UML
- Nova notação, baseada em ER
- ER

- OWL

Como se pode observar ER [17] e UML [18] são modelos de dados muito poderosos, então a maioria das soluções os envolve de alguma forma. A UML é uma ferramenta de modelagem que abrange mais situações possíveis a serem modeladas, como diagramação de casos de uso entre outras funcionalidades, por causa disso podemos observar um número significativo de casos que usam a UML em sua forma natural para modelar suas aplicações, devido a essa versatilidade.

O ER por outro lado pode parecer mais simples, porém isso pode ser observado como uma vantagem, pois isso abre espaço para criar de forma mais fácil variações do modelo que sejam melhores para modelar um determinado cenário, como é o caso do diagrama O-ER (ER-Orientado), onde os relacionamentos recebem uma notação diferente, onde são representados por arestas orientadas para melhor modelar um banco de dados de grafo, como podemos ver na Figura 9. Os números mostrados na figura são apenas uma forma que o autor usou para representar diferentes cardinalidades e não são relevantes. Na figura podemos observar um contexto de um site, que possui entidades de usuário, blog, categoria, comentário e link externo, todas conectadas por diversos relacionamentos.

Figura 9



Fonte: DE VIRGILIO, Roberto; MACCIONI, Antonio; TORLONE, Riccardo. Model-driven design of graph databases.
 In: **Conceptual Modeling: 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014.**
Proceedings 33. Springer International Publishing, 2014. p. 172-185.

Em suma podemos observar que UML e ER, assim como suas variações e modelos criados com base neles, são as soluções que mais aparecem para responder à pergunta proposta de como lidar com a modelagem conceitual de bancos NoSQL. Além dessas soluções também pôde-se observar alguns tipos de modelos que lidam com situações mais específicas, como é o caso da OWL (Web Ontology Language) [19] e, FCA (Formal Concept Analysis) [20] que são modelos criados especificamente para trabalhar com ontologias, assim como o RDF (Resource Description Framework) [21], que trata apenas de um subconjunto de bancos em grafo.

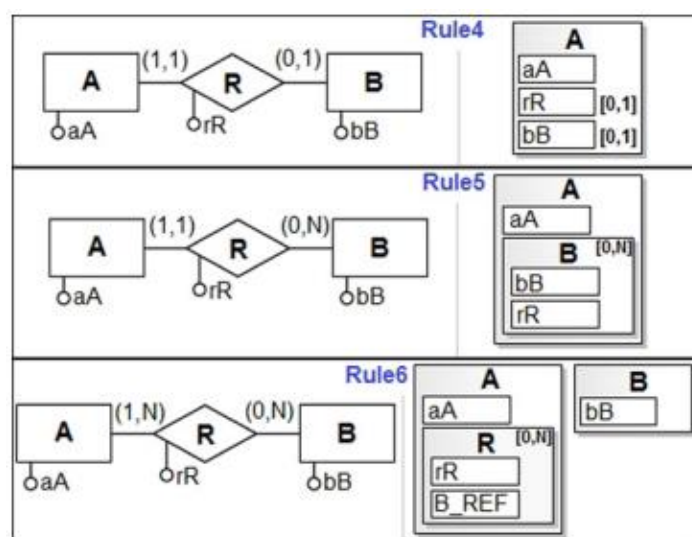
Além da pesquisa e análise de diferentes artigos foi feita uma análise de um material utilizado em um mini-curso relacionado ao projeto de banco de dados NoSQL

[31] que foi apresentado no SBBD 2022 (Simpósio Brasileiro de Banco de Dados), que tem a modelagem de bancos de dados como uma das suas principais áreas de pesquisa. O material não tem foco no modelo conceitual em si, mas sim em duas ramificações do modelo lógico, sendo uma de baixo nível e outra de alto nível, porém que tem um objetivo muito similar ao proposto nesse trabalho. A ramificação de alto nível consiste numa abstração do modelo conceitual em um modelo de dados baseado no conceito de agregado [30], que consiste na abstração de um objeto real que sua estrutura possa agregar outros objetos. Já a etapa do modelo lógico de alto nível consiste numa conversão do esquema de agregados para os modelos alvo, sejam eles chave-valor, colunar ou documentos.

Além de introduzir essas duas ramificações do modelo lógico e o conceito de agregados, o material trabalha com diversas regras de mapeamento que são divididas em duas etapas, conversão de hierarquias e conversão de relacionamentos, onde cada etapa possui diferentes abordagens que culminam em diferentes resultados. Ao fim é feita uma análise separada para bancos de dados orientados a grafos, onde são sugeridas regras de mapeamento semelhantes as propostas com base no modelo de agregados, mas que geram um modelo físico condizente com o banco de grafos.

Nessa abordagem proposta o conceito de agregados é fundamental. Tal conceito foi inicialmente proposto por Sadalage e Fowler [30] em 2012 e formalizado pelo Mello num artigo [32] de 2016. Tal conceito agrupa diferentes tipos de estruturas de dados, permitindo assim uma abstração de alto nível, o que permite que um mesmo modelo criado pela metodologia proposta possa representar qualquer tipo de banco de dados que use alguma estrutura que se encaixe na abstração. Na figura 10 temos um exemplo da conversão de entidades e relacionamentos para o modelo de agregados. Podemos observar a conversão de relacionamentos com 3 cardinalidades diferentes, que resultam por sua vez em diferentes representações no modelo de agregados, apesar de terem a mesma estrutura no modelo ER, apenas com cardinalidades distintas.

Figura 10

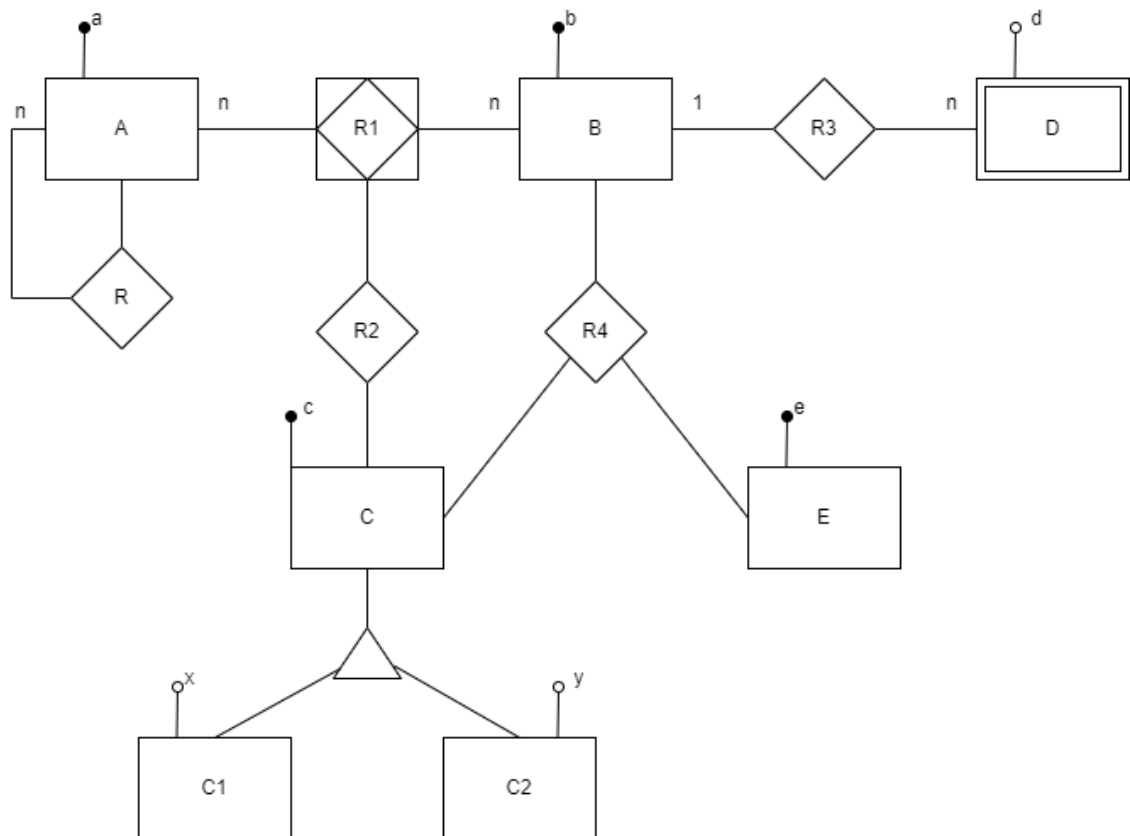


Fonte: LIMA, Cláudio; MELLO, Ronaldo Santos. On proposing and evaluating a NoSQL document database logical approach. *International Journal of Web Information Systems*, v. 12, n. 4, p. 398-417, 2016.

7. Proposta Inicial de Modelo Conceitual Poliglota

Para elaborar uma primeira proposta de modelo conceitual poliglota para os três diferentes bancos NOSQL (chave-valor, documento e grafos) foi feito um primeiro rascunho de modelos usando apenas o ER básico, sem nenhuma extensão, e também sem olhar mais nenhuma proposta de outros modelos, usando apenas os elementos básicos do ER. A escolha do ER como ferramenta para essa modelagem conceitual se deu por alguns motivos, dentre eles o fato de o ER ser a ferramenta de ensino usada nas disciplinas de banco de dados da PUC-Rio, a combinação de simplicidade e robustez que esse tipo de modelagem provê e principalmente a familiaridade do autor do relatório com o modelo. Na figura 11 temos o modelo ER que foi usado de referência para mapear os três modelos NoSQL considerados no estudo. Entre as entidades A e B temos um relacionamento n para n, ligadas por uma entidade associativa R1. Entre as entidades B e D temos o conceito de entidade fraca, ligada pelo relacionamento R3. A entidade associativa R1 se relaciona com a entidade C, que por sua vez possui uma hierarquia que gera duas especializações, C1 e C2. Por último temos um relacionamento triplo entre as entidades B, C e E.

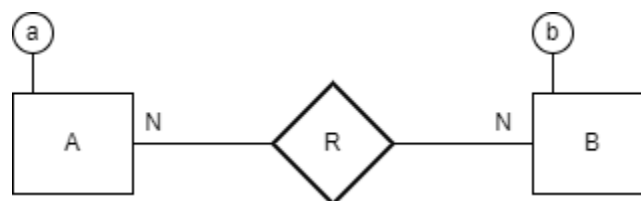
Figura 11



7.1 Do ER para Documento

Dos três tipos de bancos analisados o orientado a documento é o mais complexo, então para esse primeiro rascunho de modelo foi feita uma análise dos principais cenários que costuma ocorrer num modelo ER. O primeiro cenário, e talvez o mais importante é uma relação de duas entidades n para n , onde normalmente é criada uma nova tabela em bancos de dados relacionais, representada na figura 12.

Figura 12



Em um banco de dados orientado à documentos esse cenário pode ser resolvido de algumas formas. A primeira é com a criação de dois documentos A e B onde A possui todos os atributos de B e vice-versa, dessa forma criando uma redundância de dados que resolveria o problema de integridade. O cenário também pode ser resolvido com o

jeito mais comum em bancos relacionais, ou seja, criando um documento C que conecte todos esses dados. Além desse cenário boa parte dos problemas de um modelo ER são bem traduzidos para um banco de documentos usando o conceito de subdocumentos, ou seja, um dos atributos do documento consiste em outro documento por inteiro. Esse conceito pode ser usado para traduzir entidades fracas, hierarquias e auto relacionamentos.

Outro fator interessante de ser analisado é o de atributos multivalorados, que apesar de serem conceitualmente corretos são evitados por vezes em modelos ER pois um número significativo de SGBDs relacionais não lida bem como eles, pois as tabelas do modelo relacional as consideram incorretas devido a primeira norma formal, diferente dos documentos, que foram feitos para lidar com isso. Em suma foi observado durante a concepção de um primeiro modelo que o banco orientado a documentos pode ser modelado em um bom número de casos por um simples ER, sem a necessidade de extensões ou da criação de um tipo de modelo novo. Abaixo temos uma tabela que sintetiza o mapeamento do modelo ER para o contexto de documentos.

Tabela 1

ER	Documento
Entidade	Documento
Atributo	atributo : "atributo"
Atributo Multivalorado	atributo : [v1, v2, v3]
Conjunto de Entidades	Coleção
Relacionamento	Documento
Hierarquias	Sub Documentos
Atributo Identificador	Índices

7.2 ER -> Grafo

A mesma abordagem de tentar representar esse tipo de banco apenas com um simples ER foi usada de novo como o banco de grafos. Grafos tem algumas características importantes que precisam ser levadas em conta, são elas a orientação de um relacionamento e a forte semântica que os relacionamentos levam. A solução para a tradução de boa parte dos possíveis cenários de um ER se baseia num conceito muito utilizado em bancos relacionais, que é a criação de uma tabela intermediário que conecta os dados, nesse caso trata-se de um nó intermediário. Relacionamentos n para n como o exemplo visto no item acima pode ser traduzida com o uso de um terceiro nó

que amarra todas as informações. O modelo físico dos grafos também lida melhor com atributos multivalorados pois oferecem suporte nativo para resolver esse cenário com o uso de *arrays*, o que é outro artifício que pode ser usado para resolver alguns cenários.

Tabela 2

ER	Grafo
Entidade	Nó
Relacionamento	Aresta
Atributos	Propriedades
Atributo Identificador	Restrição de Integridade (<i>constraints</i>)
Hierarquia	Propriedades

7.3 ER -> Chave-Valor

Bancos de dados chave-valor são baseadas numa estrutura muito simples, então boa parte dos recursos que o modelo ER dispõe são muito difíceis de serem traduzidos sem nenhuma extensão. Numa primeira análise foram identificadas duas principais formas, a primeira é análoga a criação de chaves estrangeiras num banco relacional, que no caso do banco chave valor seria equivalente a, dentro do dado em si adicionar um atributo que contenha o identificador de outro dado, apesar de ser uma solução possível pela natureza caixa preta dos dados num banco chave valor, essa informação só pode ser acessada quando já se possui o identificador do dado. Outro artifício possível é a inserção de semântica dentro da chave, para essa ser buscada usando *pattern matching*. Seguindo o mesmo exemplo apresentada nos bancos orientados a documento poderia ser criado um padrão de chave [C:aaaa:bbbb] que armazenaria a identificação das duas entidades, e na parte do valor armazenar os atributos necessários.

Tabela 3

ER	Chave-Valor
Entidade, Relacionamento	Chave-Valor
Atributo Identificador	Chave
Atributos	Valor

8. Análise de Extensões

Para essa etapa do trabalho foi feita uma análise das extensões propostas pelos artigos reunidos no levantamento do artigo de referência [15]. Tal pesquisa não só reuniu artigos que tratam de modelagem conceitual, mas também de lógica e física, que não vão ser considerados. Além disso também foram desconsiderados artigos que propunham uma modelagem completamente nova, assim como modelos genéricos, logo o foco será em modelos que têm como base ER, UML e modelos específicos para ontologias. Dos 24 artigos citados no artigo de referência [15] que abordam modelagem conceitual apenas 4 tratam de SGBDs orientados a documentos, diminuindo nosso escopo de possíveis extensões que podem ser adicionadas ao modelo conceitual inicialmente proposto no item anterior. A situação piora com bancos de dados chave-valor, onde apenas um estudo foi feito no âmbito da sua modelagem conceitual, o que muito provavelmente se deve a simplicidade desse tipo de banco de dados. Após essas considerações podemos concluir que a maior parte das extensões serão aplicáveis apenas em modelos de bancos de dados orientados a grafo. Também serão analisadas as extensões vindas de 2 trabalhos que lidam com uma abordagem multi-modelo que pode ser traduzida para diferentes tipos de banco NoSQL.

8.1 Extensões para Grafos

Dentre os artigos analisados foram detectadas 3 principais abordagens para a criação de modelos conceituais de bancos orientados à grafos, dentre elas a primeira é usando o modelo ER em sua forma natural, para então traduzi-lo para uma instância de banco de grafos, como foi proposto aqui no item 7.2, logo nesses artigos [36, 37] não existe nenhuma extensão que possa ser aproveitada.

A segunda abordagem é o uso de um EB-ER, que é um modelo ER Estendido Binário, que tem algumas diferenças cruciais em relação a um ER tradicional. Tal ER binário só permite relacionamentos entre duas entidades, daí que vem o nome. O ER binário também trabalha com atributos fixos em entidades à fim de recriar o conceito de *labels* implementado pelo Neo4j, ou seja, toda entidade que possua certo *label* deve ter atributos específicos.

A terceira ferramenta utilizada na criação desses modelos, e talvez a mais interessante, foi o uso de O-ER, ER Orientados, que introduzem uma simples direção nos relacionamentos para se aproximar da linguagem utilizada por grafos direcionados. Apesar do modelo ER já ter uma direção implícita na leitura do modelo, que no caso

seria a mesma direção do sentido de leitura ocidental, da esquerda para direita, essa simples seta orientada facilita a tradução para o modelo de grafos.

8.2 Extensões para Documentos

Como previamente apontado o número de trabalhos que abordam a modelagem conceitual em bancos orientados a documentos é bem reduzido, logo nessa etapa teremos um escopo de artigos e extensões menor para analisar. Dentre as abordagens identificadas uma delas foi o uso da FCA, Formal Concept Analysis, [20] que é uma forma de descrever possíveis entidades de um banco usando uma espécie de descrição formal matemática, algo pouco visual e que foge das abordagens que vem sendo trabalhadas (ER e UML), logo esse tipo de modelagem foi desconsiderado. A outra abordagem a se notar foi o uso de um novo modelo elaborado a partir do UML, que na verdade usa os conceitos muito similares ao modelo ER só que traduzidos para um contexto de documentos, usando as mesmas estratégias elaboradas em 7.1 usando subdocumentos para traduzir diferentes tipos de cardinalidades entre objetos. Logo ao analisar os artigos que abordam os bancos orientados a documentos não foi identificada nenhuma extensão que possa ser útil para complementar a abordagem apresentada anteriormente (Proposta Inicial de Modelo Conceitual Poliglota).

8.3 Extensões para Chave-Valor

Dentre todos os artigos analisados foi identificado somente 1 que trabalha com bancos de dados chave-valor [38], que foca sua abordagem na organização dos dados dentro do campo valor, que apesar de trazer algumas possibilidades interessantes, tais informações só podem acessadas juntas da chave, componente que é pouco trabalhado no estudo. Tais bancos de dados possuem o dado em si como um componente inerentemente caixa preta, o único valor que podem ser feitas operações de busca é a chave em si. Feita essa análise se concluiu que não existe nenhuma extensão interessante, dentre os artigos avaliados pelo artigo de referência [15], que possa ser usada na modelagem de bancos chave valor e a abordagem a ser seguida será a de inserção de semântica nas chaves por meio de diferentes técnicas.

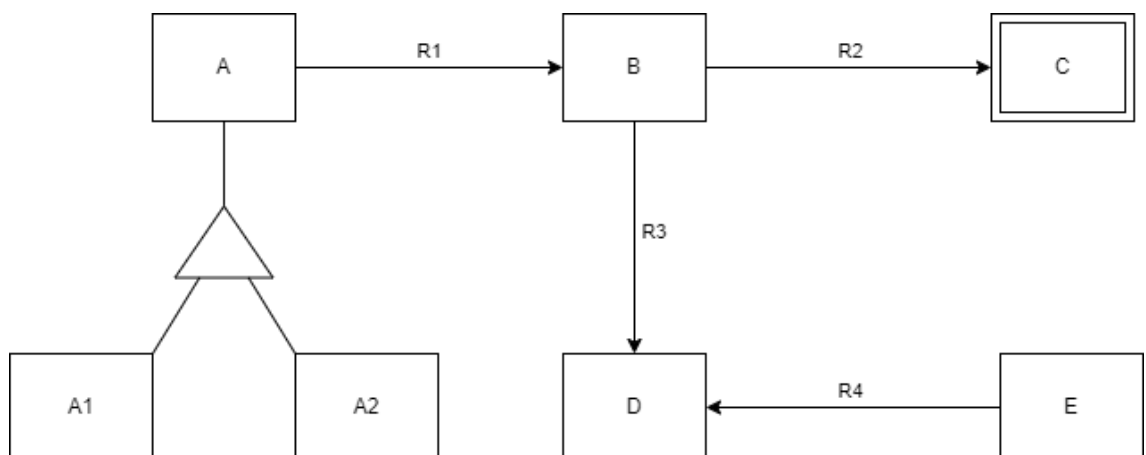
9. Proposta de Modelo Conceitual Poliglota

Após a proposta inicial e a análise de extensões foi proposto um modelo conceitual, a partir de duas extensões realizadas do modelo ER (identificadas na seção 8), assim como diversas regras de conversão. O modelo poderá ser usado de algumas formas diferentes, tendo o foco dele em um tipo de banco específico, ou sendo tratado como multi modelo, de forma que possa contemplar mais de um tipo de banco diferente em um só modelo.

9.1 Foco em Grafo

Para melhor definir o conceito de grafos no modelo ER, usaremos o conceito de ER Orientado, visto na análise de extensões para grafos. Importante ressaltar que apesar dos relacionamentos terem uma notação diferente no formato de setas orientadas eles ainda comportam o uso de atributos.

Figura 13



Também seguiremos algumas regras de conversão para transformação do modelo:

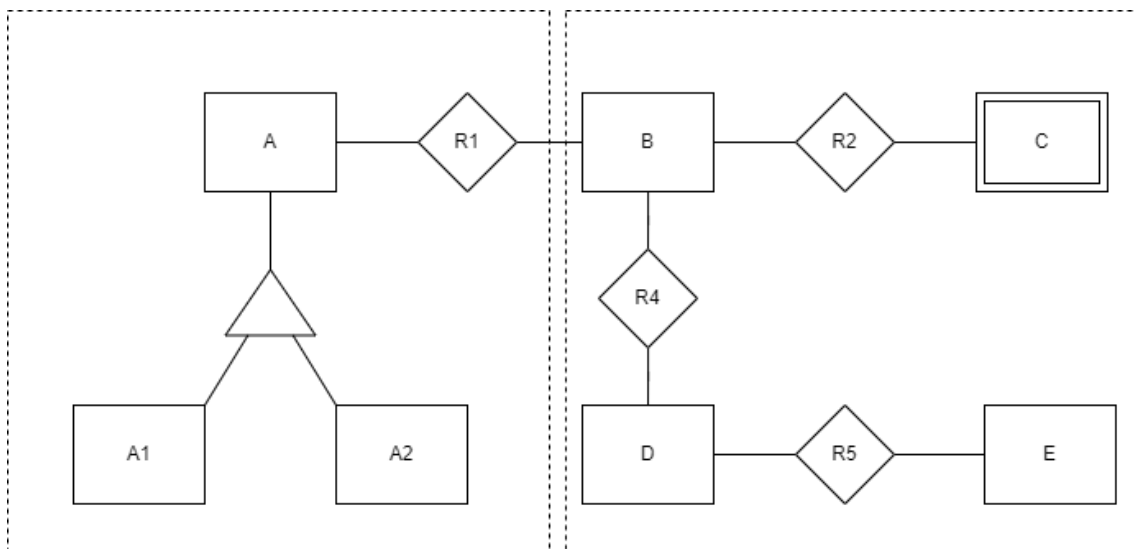
1. Em relacionamentos 1-1 geram 2 nós individuais, onde um nó recebe a chave do outro. Em alguns casos ambos os nós podem receber a chave do outro se redundância não for um problema.
2. Em relacionamentos 1-n o nó de cardinalidade n recebe o atributo identificador do outro nó. Grafos não tem id, esse atributo seria gerado de forma artificial.
3. Em relacionamentos n-n será feita a criação de um terceiro nó intermediário.
4. Em entidades fracas usaremos o conceito de aninhamento, a entidade será armazenada num atributo multivalorado do nó principal.

5. Em especializações será usado o mesmo princípio, onde as entidades que dependem da principal serão armazenadas como um atributo dentro do nó.
6. Em modelos que sejam feitos especificamente para o Neo4J os *labels* (rótulos) serão armazenadas no ER como um atributo multivalorado chamado *labels*.
7. Atributos identificadores podem ter sua integridade mantida usando restrições providas pelo SGBD de escolha.

9.2 Foco em Documentos

Para melhor definir o modelo para um banco de documentos usaremos o ER tradicional, mas iremos adicionar uma notação especial para explicitar melhor a divisão entre coleções.

Figura 14



Também seguiremos algumas regras de conversão para transformação do modelo:

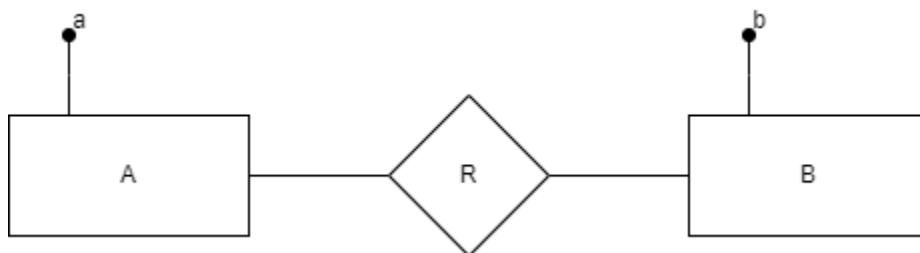
1. Em relacionamentos 1-1 serão gerados 2 documentos distintos, onde um deles recebe o “_id” do outro. Em alguns casos ambos os nós podem receber a chave do outro se redundância não for um problema.
2. Em relacionamentos 1-n o documento de cardinalidade n recebe o “_id” do outro documento.
3. Em relacionamentos n-n criamos um terceiro documento que liga as informações das duas entidades originais.

4. Entidades fracas são tratadas como subdocumentos, a entidade dependente vira um atributo dentro da entidade forte.
5. Em especializações se aplica o mesmo conceito de subdocumentos, onde as entidades dependentes viram um atributo dentro do documento pai.
6. Diferentes coleções dentro do modelo serão separadas em retângulos pontilhados, assim facilitando a criação do modelo lógico e físico.
7. Atributos identificadores podem ser armazenados em índices que garantem sua individualidade e são providos pelos SGBDs

9.3 Foco em Chave Valor

Como já estabelecido várias vezes, esse tipo de banco é bem mais limitado, então não será aplicada nenhuma extensão ao modelo ER e usaremos apenas regras de conversão para a criação de chaves.

Figura 15



1. Em relacionamentos 1-1 será criada uma chave simples [A:aaaa] e a chave do relacionamento B será referenciado dentro do campo valor, e só será acessado tendo a chave A, e o mesmo será feito com a entidade B, para a busca poder ser efetuada tendo qualquer um dos valores
2. Em relacionamentos 1-n a entidade de cardinalidade n terá uma chave simples [A:aaaa] e o id do relacionamento B será referenciado dentro do campo valor, e só será acessado tendo a chave A.
3. Em relacionamentos n-n será criada uma chave que modelo o relacionamento R como um todo [R:aaaa:bbbb] que guarda a chave do relacionamento A e B respectivamente

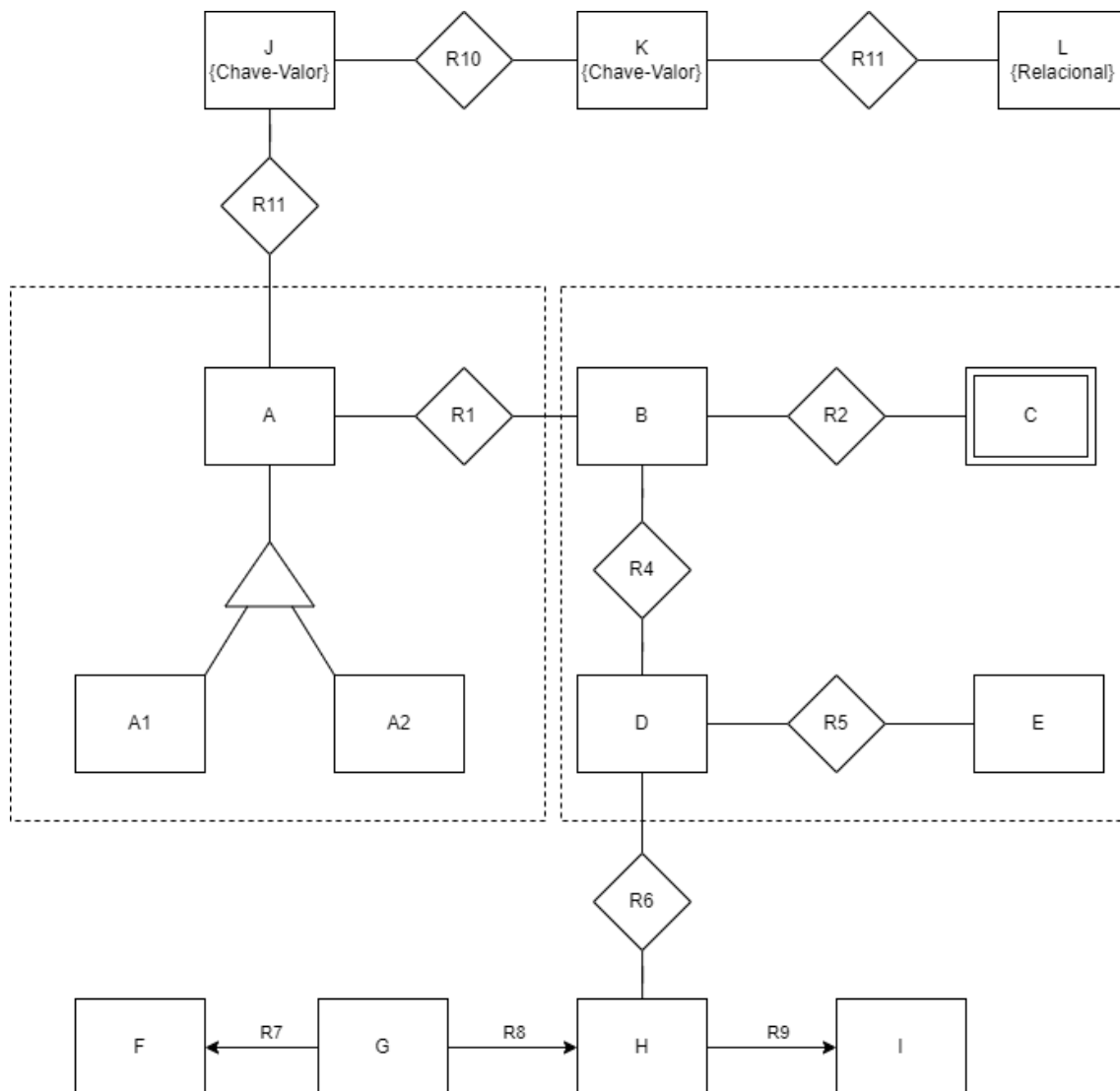
9.4 Multi Modelo

Todas as regras dispostas acima podem ser implementadas num único modelo de forma que diferentes seções dele serão traduzidas para diferentes tipos de bancos.

Essa divisão pode ser feita de diversas formas, pelo uso de cores que representam diferentes modelos lógicos de dados, fazendo o uso de um seccionamento gráfico com caixas ou linhas que englobem diferentes bancos, mas com esse método pode-se gerar confusão com a divisão de coleções. Por último pode ser feita a criação de um rótulo em todas as entidades que represente para qual tipo de banco aquela entidade será mapeada. Tal *label* serve principalmente para explicitar a estrutura de dados de entidades onde não existam extensões que evidenciem isso, como é o caso do grafo com os relacionamentos direcionados, e dos documentos, como é o caso da divisão de coleções.

Nesta proposta definitiva de um modelo poliglota foi utilizada uma extensão de relacionamentos direcionados, para facilitar o entendimento dos dados que no futuro serão armazenados num banco de grafos, essa extensão surgiu da seção 8 do trabalho de análise de extensões. Já a divisão de coleções por meio de um seccionamento pontilhado, assim como os labels em entidades onde não é explícito para qual tipo de banco aquela entidade será armazenada, são extensões de autoria própria, desenvolvidas ao longo do trabalho.

Figura 16



10. Validação do Modelo Proposto

Após a segunda proposta do modelo, complementado pelas extensões analisadas, foi feita uma validação do que havia sido feito, como meio de se avaliar o modelo, assim como sua eficácia em modelar o que foi proposto, bancos de dados de grafos, documentos e chave-valor. Para isso foi feito o contato com diversos professores e professoras da PUC-Rio que pesquisem ou trabalhem com modelagem de dados. Foram feitas diversas tentativas de contato com alguns professores que não foram sucedidas, porém houve um contato, assim como uma conversa, bem-sucedida com a professora Fernanda Baião, do Departamento de Engenharia Industrial da PUC-Rio, que tem entre suas principais áreas de pesquisa a modelagem conceitual. O feedback da professora tinha como principal ponto o fato que ao olhar a base da teoria da

modelagem de dados conceitual, não faz sentido um modelo representar a estrutura de dados em que ele será implementado no futuro. De fato, o modelo conceitual, no caso do ER, modelo escolhido nesse trabalho, foi concebido para ser mapeado para bancos de dados hierárquicos, em redes e relacional, de forma que em nenhum ponto do modelo fosse refletido o modelo lógico que aquele modelo conceitual serviria. Um ponto importante a se notar é que no momento da concepção do ER não era considerado um projeto lógico do multi-modelo, pois até então não existiam SGBDs que trabalhassem com mais de um tipo de banco de dados de forma simultânea, assim como era incomum aplicações fazerem uso de diferentes tipos de bancos de dados.

Esse feedback inicialmente parece apontar uma falha grave no cerne desse trabalho, mas tudo pode ser resolvido de forma simples, com uma pequena alteração. Mantendo todas as extensões propostas, assim como regras de mapeamento o modelo proposto, ao invés de se tratar de um modelo conceitual, passaria a atuar numa camada intermediária entre o modelo conceitual e o modelo lógico, de forma que essa etapa simplifique a distinção multi-modelo, assim resultando numa melhor concepção do modelo lógico.

Em suma se conclui com o feedback da professora Fernanda Baião, que o ER de fato pode modelar bancos de dados em grafos, documentos e chave-valor sem a necessidade de extensões, porém em situações multi-modelo é introduzida o modelo proposto numa camada extra entre a camada conceitual e lógica a fim de melhor representar situações que envolvam mais de um banco de dados envolvido em um mesmo modelo. Na figura 16 temos uma visualização gráfica de onde essa camada seria inserida na modelagem de banco de dados.

Figura 17

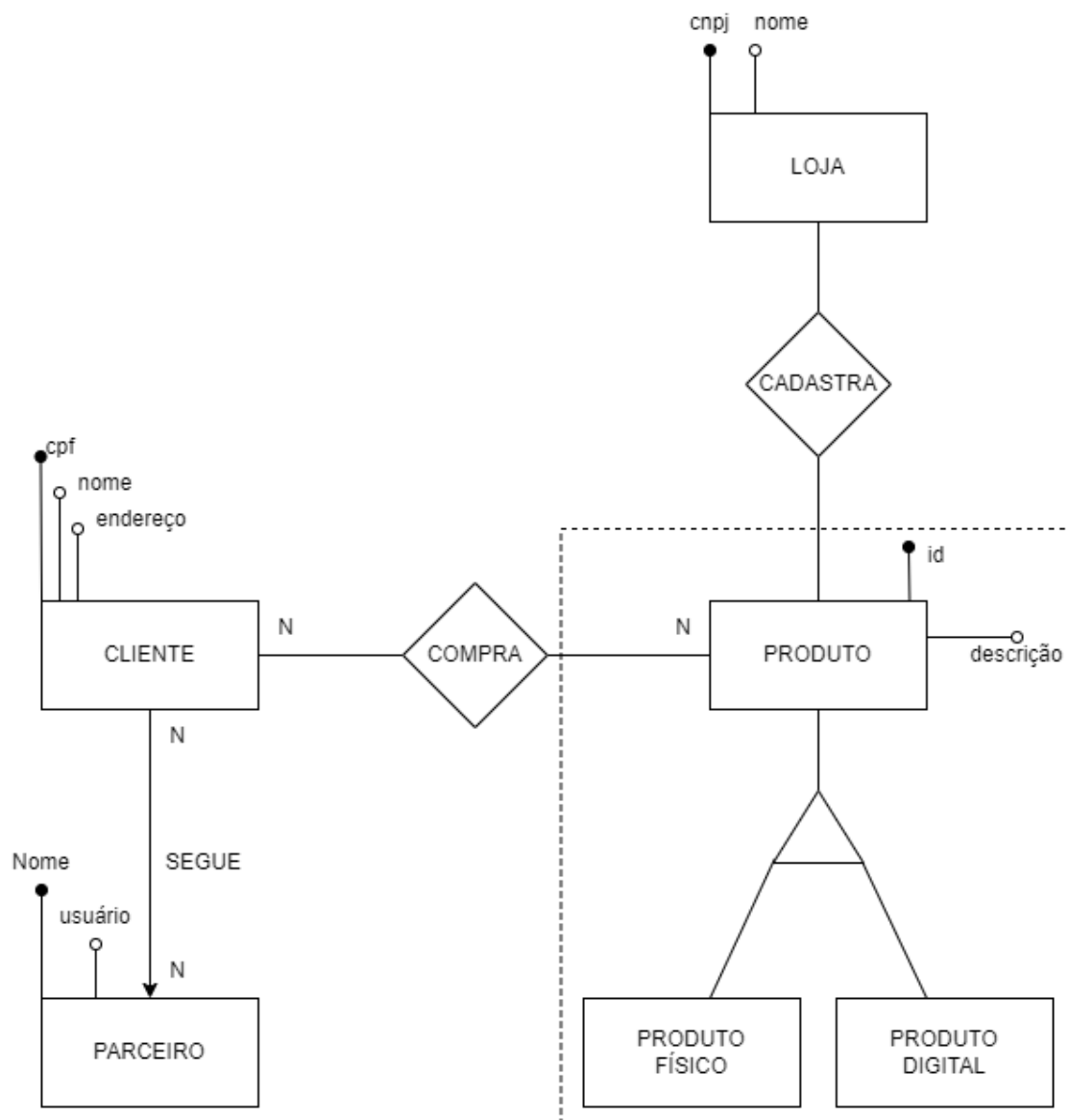


11. Exemplo de Aplicação

Nesse tópico será proposto um sistema fictício que ilustre todas as ferramentas de modelagem estudadas e elaboradas ao longo do projeto. Não se trata de uma proposta de um sistema eficiente ou otimizado, e sim apenas uma ferramenta para ilustra um caso de uso da proposta definitiva de um modelo poliglota.

Uma empresa de e-commerce decidiu criar uma nova plataforma e usar diferentes tipos de bancos de dados para armazenar seus dados. Nessa plataforma um cliente pode comprar vários produtos que tem um grande estoque. As informações dos clientes são sempre iguais e serão armazenadas num banco de dados relacional. Os produtos podem ser tanto físicos quanto digitais e cada produto pode conter diversas informações diferentes e serão armazenados num banco de dados de documentos, numa única coleção. A plataforma contrata influenciadores para serem seus parceiros e os cliente podem segui-los por meio da plataforma. Esse aspecto social da plataforma será modelado num banco de dados de grafo. Na plataforma diferentes lojas podem cadastrar diferentes produtos. Essa informação será armazenada num banco de dados chave valor, usando uma chave que junte a identificação da loja e do produto no formato [C:LLLL:PPPP]

Figura 18



12. Conclusão

Com base na coleta de artigos e no estudo do estado da arte da modelagem conceitual de dados, pudemos constatar que não existe um consenso em como gerar modelos conceituais para bancos NoSQL. Revendo os objetivos do projeto tínhamos como meta a proposta de um modelo que pudesse contemplar diferentes tipos de banco de dados NoSQL, meta essa que foi alcançada ao longo da elaboração do trabalho. O modelo elaborado usa simples extensões que ajudam a melhor especificar certos modelos lógicos, mas ele é baseado no modelo ER, logo ele ainda comporta mais

extensões em cima dele, e pode ser trabalhado com base em diferentes bancos de dados que possam vir a surgir, ou que façam sentido numa aplicação específica. Em suma foi gerado um modelo simples e versátil que age numa camada intermediária entre o modelo conceitual e lógico e que ainda permite espaço para futuras alterações e modificações.

Pensando em possíveis continuações futuras para o trabalho vale notar algumas das principais ideias. Em primeiro lugar e talvez a mais direta dessas continuações é a inclusão de outros diferentes tipos de bancos de dados NoSQL a serem contemplados pelo projeto. Essa área ainda se trata de algo novo na ciência da computação e cada vez mais vem surgindo novos tipos de bancos para trabalhar com novas tecnologias, então sempre será interessante atualizar o projeto de forma que ele englobe e seja capaz de modelar novas tecnologias que venham a surgir no futuro.

Em segundo lugar podemos pensar numa extensão do trabalho para o modelo UML. Apesar do foco do trabalho no modelo ER a UML também é amplamente usada, logo seria interessante idealizar equivalências para essa linguagem com base no que foi desenvolvido ao longo do projeto para o ER. Dessa forma seria possível que na criação do projeto de banco de dados os desenvolvedores possam escolher o tipo de modelagem que seja mais familiar para eles, mas ainda tendo a capacidade de modelar todas as situações propostas nesse trabalho.

O projeto também possui diversas futuras aplicações, mas agora no campo de software. Já existem diversas ferramentas de modelagem de dados virtuais, desde as mais simples e genéricas, como o draw.io [39], até ferramentas mais específicas para a modelagem de banco de dados, como é o caso do BrModelo [40], projeto desenvolvido pelo professor Ronaldo dos Santos Mello, da Universidade Federal de Santa Catarina. Tais ferramentas poderiam receber extensões ou modificações que permitiriam a modelagem das extensões propostas aqui, assim como a sugestão de suas regras de mapeamento.

13. Referências Bibliográficas

[1] ELMASRI, Ramez e NAVATHE, Shamkant B. Sistemas de Banco de Dados. Pearson Addison Wesley. 7a Edição, 2019.

[2] Codd, Edgar F. *Relational completeness of data base sublanguages*. IBM Corporation, 1972.

[3] DB-ENGINES. DB-Engines Database Ranking. Disponível em <<https://db-engines.com/en/ranking>>. Acesso em 20 de março de 2023.

[4] Imam, Abdullahi Abubakar, et al. "Schema proposition model for NoSQL applications." *Recent Trends in Data Science and Soft Computing: Proceedings of the 3rd International Conference of Reliable Information and Communication Technology (IRICT 2018)*. Springer International Publishing, 2019.

[5] Berg, Kristi L., Tom Seymour, and Richa Goel. "History of databases." *International Journal of Management & Information Systems (IJMIS)* 17.1 (2013): 29-36.

[6] Batra, Rahul, and Rahul Batra. "A history of SQL and relational databases." *SQL Primer: An Accelerated Introduction to SQL Basics* (2018): 183-187.

[7] Scherzinger, Stefanie, and Sebastian Sidortschuck. "An empirical study on the design and evolution of NoSQL database schemas." *Conceptual Modeling: 39th International Conference, ER 2020, Vienna, Austria, November 3–6, 2020, Proceedings* 39. Springer International Publishing, 2020.

[8] Borgida, A., Mylopoulos, J. (2009). Conceptual Schema Design. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA.

[9] Atzeni, Paolo, et al. "Data modeling in the NoSQL world." *Computer Standards & Interfaces* 67 (2020): 103149.

[10] Redis. What is a Key-Value Database? Disponível em <<https://redis.com/wp-content/uploads/2020/10/key-value-figure-b-v2-1024x314.png?auto=webp&quality=85,75&width=1200>>

[11] Wikipedia. Graph Databases. Disponível em <https://upload.wikimedia.org/wikipedia/commons/3/3a/GraphDatabase_PropertyGraph.png>

[12] Neo4J. Cypher Query Language. Disponível em <<https://dist.neo4j.com/wp-content/uploads/sample-cypher.png>>

[13] MongoDB. What is a Document Database. Disponível em <https://webimages.mongodb.com/_com_assets/cms/kq6uev0p1jnpq1ob3-snippet_light.svg?auto=format%2Ccompress&ch=DPR>

[14] HENRICSSON, Robin. Document Oriented NoSQL Databases: A comparison of performance in MongoDB and CouchDB using a Python interface. 2011.

- [15] VERA-OLIVERA, Harley et al. Data modeling and NoSQL databases-a systematic mapping review. *ACM Computing Surveys (CSUR)*, v. 54, n. 6, p. 1-26, 2021.
- [16] DE VIRGILIO, Roberto; MACCIONI, Antonio; TORLONE, Riccardo. Model-driven design of graph databases. In: **Conceptual Modeling: 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014. Proceedings 33**. Springer International Publishing, 2014. p. 172-185.
- [17] CHEN, Peter Pin-Shan. The entity-relationship model—toward a unified view of data. **ACM transactions on database systems (TODS)**, v. 1, n. 1, p. 9-36, 1976.
- [18] UML. What is UML. Disponível em: <<https://www.uml.org/what-is-uml.htm>>
- [19] W3C. Web Ontology Language. Disponível em <<https://www.w3.org/OWL/>>
- [20] WILLE, Rudolf; GANTER, Bernhard. Formal concept analysis. 1996.
- [21] W3C. Resource Description Framework. Disponível em <<https://www.w3.org/RDF/>>
- [23] BACHMAN, Charles W. Data structure diagrams. **ACM SIGMIS Database: The DATABASE for Advances in Information Systems**, v. 1, n. 2, p. 4-10, 1969.
- [24] IBM. Advantages of XML. Disponível em: <<https://www.ibm.com/docs/en/i/7.3?topic=introduction-advantages-xml>>
- [25] Oracle. JSON Defined. Disponível em: < <https://www.oracle.com/database/what-is-json/> >
- [26] ScienceDirect. Semistructured Data. Disponível em <<https://www.sciencedirect.com/topics/computer-science/semistructured-data>>
- [27] Graphviz. DOT Language. Disponível em < <https://www.graphviz.org/doc/info/lang.html> >
- [28] Neo4J. Use Cases. Disponível em: < <https://neo4j.com/use-cases/> >
- [29] Amazon AWS. O que é um banco de dados de grafos. Disponível em < <https://aws.amazon.com/pt/nosql/graph/> >
- [30] SADALAGE, P. J.; FOWLER, M. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**. EUA, 2012
- [31] MELLO, Ronaldo Santos. Minicurso: Projeto de Bancos de Dados NoSQL. 2022. Apresentação de Power Point.

- [32] LIMA, Cláudio; MELLO, Ronaldo Santos. On proposing and evaluating a NoSQL document database logical approach. **International Journal of Web Information Systems**, v. 12, n. 4, p. 398-417, 2016.
- [33] VILLA, Fernán; MORENO, Francisco; GUZMÁN, Jaime. An Analysis of a Methodology that Transforms the Entity-Relationship Model into a Conceptual Model for a Graph Database. In: **Emerging Technologies in Computing: First International Conference, iCETiC 2018, London, UK, August 23–24, 2018, Proceedings 1**. Springer International Publishing, 2018. p. 70-83.
- [34] DE VIRGILIO, Roberto; MACCIONI, Antonio; TORLONE, Riccardo. Model-driven design of graph databases. In: **Conceptual Modeling: 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014. Proceedings 33**. Springer International Publishing, 2014. p. 172-185.
- [35] VARGA, Viorica; JÁNOSI-RANCZ, Katalin Tünde; KÁLMÁN, Balázs. Conceptual design of document NoSQL database with formal concept analysis. **Acta Polytech. Hungarica**, v. 13, n. 2, p. 229-248, 2016.
- [36] POKORNÝ, Jaroslav. Conceptual and database modelling of graph databases. In: **Proceedings of the 20th international database engineering & applications symposium**. 2016. p. 370-377.
- [37] VÁGNER, Anikó. Store and visualize EeR in Neo4j. In: **Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control**. 2018. p. 1-6.
- [38] BUGIOTTI, Francesca et al. Database design for NoSQL systems. In: **Conceptual Modeling: 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014. Proceedings 33**. Springer International Publishing, 2014. p. 223-231.
- [39] Drawio. About. Disponível em: < <https://github.com/jgraph/drawio#readme> >
- [40] brModelo. Projeto BrModelo 3.0. Disponível em: < <http://www.sis4.com/brModelo/> >
- [41] MELLO, Ronaldo Santos. Persistência Poliglota: Fundamentos, Desafios e Oportunidades. 2022. Apresentação de Power Point.