

## Critérios para Seleção de SGBD NoSQL: o Ponto de Vista de Especialistas com base na Literatura

Alexandre Morais de Souza<sup>1</sup>, Edmir P. V. Prado<sup>1</sup>, Violeta Sun<sup>1</sup>, Marcelo Fantinato<sup>1</sup>

<sup>1</sup>Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)  
Rua Arlindo Bétio, 1000 – CEP: 03828-000 – São Paulo – SP – Brazil

{alexandre.souza, eprado}@usp.br

**Abstract.** *The theme of Big Data has grown in importance in the context of web applications. As a consequence, NoSQL DBMSs emerged in order to achieve gains in performance and scalability. Within this context, this study aims to identify the main criteria for selecting a NoSQL DBMS in private organizations. This research identified through literature review the main criteria for selection of NoSQL DBMSs. Those criteria were checked by a group of 32 specialists and as a result, 22 criteria are identified to be followed by organizations in their selection processes.*

**Resumo.** *O tema do Big Data cresceu em importância no contexto das aplicações web e como consequência surgiram os SGBDs NoSQL com o objetivo de obter ganhos de desempenho e escalabilidade. Dentro desse contexto, esta pesquisa teve como objetivo identificar os principais critérios para seleção de um SGBD NoSQL em organizações privadas. A pesquisa identificou na literatura os principais critérios de seleção de SGBDs NoSQL e os verificou junto a um grupo de 32 especialistas no tema. Como resultado, foram identificados 22 critérios que devem ser observados pelas organizações em seus processos de seleção.*

### 1. Introdução

O grande volume de dados gerado por aplicações *web*, juntamente com requisitos diferenciados como a escalabilidade e o elevado grau de disponibilidade, têm contribuído para o surgimento de novos modelos e tecnologias com relação aos SGBDs (Sistemas Gerenciadores de Banco de Dados). As redes sociais, por exemplo, requerem o gerenciamento de grandes quantidades de dados não estruturados que são gerados diariamente por milhões de usuários.

Neste ambiente de crescente demanda por volume de dados e desempenho, surge um novo modelo de SGBD, chamado NoSQL, ou seja, não apenas SQL (Linguagem de Consulta Estruturada). Este novo modelo foi proposto com o objetivo de atender ao gerenciamento de grandes volumes de dados semiestruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade [Lóscio *et al.* 2011]. A necessidade de um novo conceito em banco de dados surgiu como consequência da ineficiência de SGBDs relacionais em lidar com o atual volume de informações das aplicações *web*. Os SGBDs relacionais foram propostos na década de 70, quando as aplicações lidavam apenas com dados estruturados, ou seja, dados que possuem uma estrutura fixa e bem definida. Além disso, o volume de dados gerado por tais aplicações é muito menor do que o volume gerado pelas aplicações *web* atuais.

Surgem então diferentes tipos de SGBDs NoSQL, com diferentes abordagens, mas possuindo a semelhança de serem não relacionais. Eles possuem esta denominação, pois armazenam os dados de forma não estruturada e não exigem esquemas fixos como os SGBD relacionais, onde em muitos casos não oferecem apoio a operações de junção como no modelo relacional [Leavitt 2010].

A motivação deste trabalho se deve aos poucos estudos relacionados com processos de seleção de SGBDs NoSQL. Esta pesquisa tem a finalidade de ajudar a preencher essa lacuna. Dentro desse contexto, esta pesquisa tem como objetivo geral identificar os principais critérios que devem ser usados para seleção de um SGBD NoSQL em organizações privadas. Para consecução desse objetivo geral foram identificados dois objetivos específicos: (1) identificar na literatura os critérios de seleção de SGBDs; e (2) consolidar esses critérios junto a especialistas em SGBDs NoSQL.

## 2. SGBDs NoSQL

Os SGBDs NoSQL não fazem uso da álgebra relacional e nem possuem linguagem de consulta nativa SQL, o que torna mais complexa a atividade de desenvolvimento. Normalmente os SGBDs NoSQL possuem estrutura simplificada, sem estrutura de relacionamentos, com suporte natural a replicação. São também conhecidos como banco de dados escaláveis e direcionados para aplicações *web*, devido sua característica inerente de ter bom desempenho com grande volume de dados.

Os SGBDs NoSQL apresentam algumas características fundamentais que os diferenciam dos tradicionais SGBDs relacionais, tornando-os adequados para armazenamento de grandes volumes de dados não estruturados ou semiestruturados. Segundo Lóscio *et al.* (2011), é possível descrever algumas destas características abaixo:

**a) Escalabilidade horizontal.** A ausência de controle de bloqueios é uma característica dos SGBDs NoSQL torna esta tecnologia adequada para solucionar problemas de gerenciamento de volumes de dados que crescem exponencialmente, como os dados da *web 2.0*.

**b) Ausência de esquema ou esquema flexível.** Uma característica evidente dos SGBDs NoSQL é a ausência completa ou quase total do esquema que define a estrutura dos dados modelados. Esta ausência facilita tanto a escalabilidade quanto contribui para um maior aumento da disponibilidade. Em contrapartida, não há garantias da integridade dos dados, o que ocorre nos bancos de dados relacionais, devido à sua estrutura rígida.

**c) Permite replicação de forma nativa.** Diminui o tempo gasto para recuperar informações. Existem duas abordagens principais para replicação: *Master-Slave* (Mestre-Escravo) e *Multi-Master*.

**d) Consistência eventual.** A consistência nem sempre é mantida entre os diversos pontos de distribuição de dados. Esta característica tem como princípio o teorema CAP (*Consistency, Availability and Partition Tolerance*), que diz que, em um dado momento, só é possível garantir duas de três propriedades entre consistência, disponibilidade e tolerância à partição [Ye e Li 2011].

Lóscio *et al.* (2011) classificaram os principais tipos de modelos de dados NoSQL em: chave-valor, orientado a colunas, orientado a documentos e orientado a grafos. Para esses autores um determinado modelo não deve ser considerado melhor que o outro, pois cada modelo é mais adequado para determinados contextos das aplicações.

## 2.1. Chave-Valor

O modelo chave-valor (*key-value*) é considerado simples e permite a visualização do banco de dados como uma grande tabela *hash*, que é uma estrutura de dados especial que associa chaves de pesquisa a valores. De maneira bem simples, o banco de dados é composto por um conjunto de chaves, as quais estão associadas a um único valor.

Essa estrutura de dados é extremamente popular porque proporciona grande eficiência. Um par chave-valor é um valor único que pode ser facilmente utilizado para acessar os dados [Tiwari 2011]. Este modelo é de fácil implementação, permite que os dados sejam rapidamente acessados pela chave e contribui para aumentar a capacidade de acesso aos dados principalmente em sistemas que possuem alta escalabilidade.

Na Figura 1 há uma representação da estrutura chave-valor com um exemplo de aplicação para *e-commerce*. Este é um típico exemplo onde se pode combinar o uso do SGBD NoSQL orientado a chave-valor e o modelo relacional.

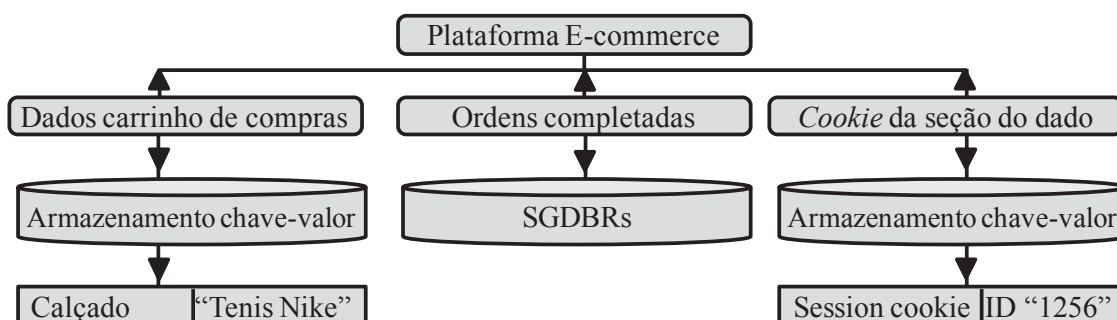


Figura 1. Banco de Dados NoSQL com abordagem Chave-Valor  
Fonte: Adaptado de Sadalage, Fowler e River (2013)

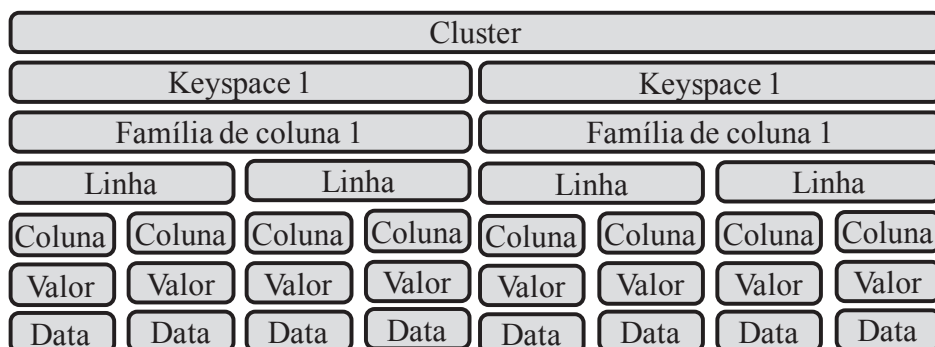
## 2.2. Orientado a Colunas

O modelo orientado a coluna é mais complexo que o modelo chave-valor [Pokorny 2011]. Neste modelo os dados são indexados por uma tripla (linha, coluna e *timestamp*), onde linhas e colunas são identificadas por chaves e o *timestamp* permite diferenciar múltiplas versões de um mesmo dado. Outro conceito associado ao modelo é o de família de colunas (*column family*), que é usado com o intuito de agrupar colunas que armazenam o mesmo tipo de dados.

Segundo Tiwari (2011), os bancos orientados a colunas são os mais populares entre os bancos não relacionais. Embora os bancos de dados orientados a colunas existem de diferentes formas há alguns anos, eles foram popularizados para a comunidade de desenvolvedores com as publicações de artigos do Google© [Ghemawat, Gobioff e Leung 2003; Dean e Ghemawat 2008; Chang, Dean e Ghemawat 2008].

Apesar dos SGBDs orientados a colunas serem semelhantes ao modelo relacional, na verdade eles são completamente diferentes. Não se pode aplicar o mesmo

tipo de solução utilizado na forma relacional para um banco de dados orientado a colunas [Rahien 2010]. A Figura 2 apresenta um exemplo desse modelo.



**Figura 2. Modelo Família de Coluna**  
**Fonte: Adaptado de Sadalage, Fowler e River (2013)**

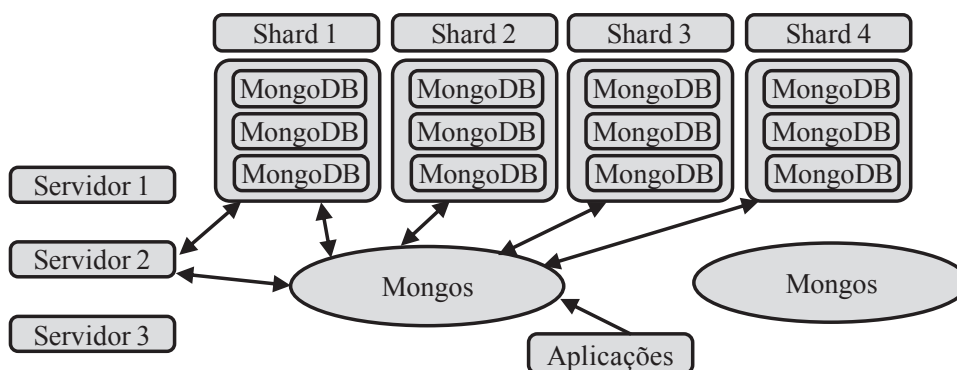
### 2.3. Orientado a Documentos

O modelo orientado a documentos armazena coleções de documentos. No modelo chave-valor apenas uma única tabela *hash* é criada para todo o banco. No modelo orientado a documentos existe um conjunto de documentos e em cada documento há um conjunto de campos (chaves) e o valor deste campo [Pokorny 2011].

As bases de dados orientados a documentos possuem o foco em resolver problemas causados por bancos de dados relacionais. Trata-se do esforço necessário para traduzir objetos de um SGBDR, o que se torna um empecilho para a produtividade do programador quando surge a necessidade de alterar alguma característica da modelagem de dados. Existem ferramentas, como Hibernate, para automatizar esse mapeamento, mas apenas resolvem parcialmente o problema. A modelagem de dados em bancos orientados a documentos são mais flexíveis quanto a alterações. Nos bancos relacionais qualquer tipo de alteração no modelo de dados é custoso, pois os programas precisam ser modificados e compilados em conjunto com a alteração do modelo. Em bases de dados maiores, alteração no modelo envolve na propagação de alterações em vários outros bancos compartilhados. Nas bases de dados orientados a documento, uma aplicação pode modificar a estrutura do documento quanto quiser sem impactos para aplicação, com o custo de riscos de inconsistências ou a existência de documentos obsoletos [Harrison 2010].

Na Figura 3 é apresentado o funcionamento da estrutura de *sharding*, onde os dados são divididos em vários *shards*, cada um contando com sua própria estrutura de replicação.

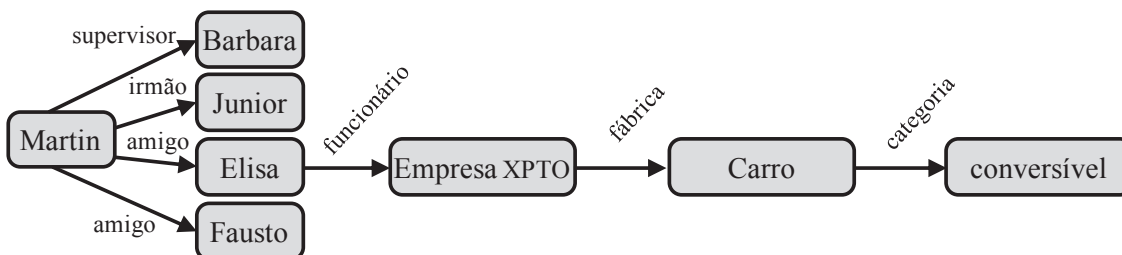
Numa base de dados orientada a documentos, o documento pode mapear quase que diretamente para linguagem de estrutura de classes. Isto faz com que a programação seja mais fácil, mas faz levantar questões de integridade de dados, uma vez que alguns itens de dados são quase inevitavelmente duplicados [Harrison 2010].



**Figura 3. Estrutura de Sharding**  
 Fonte: Adaptado de Tiwari (2011)

### 2.4. Orientado a Grafos

O modelo orientado a grafos possui três componentes básicos: os nós (vértices do grafo), os relacionamentos (as arestas) e as propriedades dos nós e relacionamentos. Este modelo permite o armazenamento das entidades e dos relacionamentos entre as elas. Assim, pode-se armazenar o dado de uma forma e interpretá-lo de diferentes formas baseado no seu relacionamento. A Figura 4 ilustra um exemplo desse tipo de modelo.



**Figura 4. Banco de Dados NoSQL abordagem com Grafos**  
 Fonte: Adaptado de Sadalage, Fowler e River (2013)

## 3. Método da Pesquisa

### 3.1. Definição do Problema de Pesquisa

Segundo Elmasri e Navathe (2005), a escolha de um SGBD é regida por vários critérios: alguns técnicos, outros econômicos e outros ainda ligados à política da organização. Os critérios técnicos se preocupam com a conveniência do SGBD em relação à tarefa que se tem. Aspectos a considerar: o tipo de SGBD (relacional, NoSQL, objeto-relacional, objeto, hierárquico, entre outros), a estrutura de armazenamento e os caminhos de acesso que o SGBD suporta, as interfaces disponíveis para usuários e programadores, os tipos de linguagens de consulta de alto nível, a disponibilidade de ferramentas de desenvolvimento, a habilidade para conectar-se a outros SGBDs por meio de interfaces padrão, as opções arquitetônicas relacionadas a operações cliente-servidor e assim por diante. Critérios não técnicos incluem a situação financeira e a estrutura de suporte do fornecedor. No entanto deve-se também considerar outros fatores como: a forma de distribuição e comercialização, qualificação dos profissionais envolvidos, estrutura de treinamento da organização, cultura organizacional, recursos, infraestrutura etc.

O foco deste trabalho são os critérios de seleção de SGBDs NoSQL. Para o levantamento desses critérios foi realizada uma revisão sistemática da literatura.

### 3.2. Revisão Sistemática

Os trabalhos científicos considerados na revisão sistemática envolvem projetos sobre SGBDs NoSQL, comparação entre SGBDs, e critérios de seleção de SGBDs. As fontes utilizadas foram consultadas em bases de dados que disponibilizam artigos completos da área de SI, com especial interesse na área de banco de dados.

As buscas foram realizadas utilizando máquinas de busca de editores ou bibliotecas digitais disponíveis. As bases de dados consultadas foram: (1) IEEE Digital Library (<http://ieeexplore.ieee.org/Xplore>); (2) ACM Digital Library (<http://portal.acm.org/>); (3) Portal de Periódicos da Cape (<http://novo.periodicos.capes.gov.br>); e (4) Scopus Database Literature (<http://www.scopus.com/>). Nessas bases de dados foram feitas pesquisas com as seguintes palavras-chave: NoSQL e *database*; NoSQL, *database* e DBMS; *Evaluate* e DBMS; e *Select* e DBMS. Foram considerados os artigos relacionado a SGBDs NoSQL publicados nos últimos cinco anos.

No total foram encontrados 38 documentos. Desses documentos, foram escolhidos 15 artigos para realização de leitura minuciosa e elaboração de resumos, pois traziam a temática de SGBDs NoSQL e tratavam sobre processo de seleção de SGBDs. Após a análise dos 15 artigos foram selecionados 12, conforme ilustra a Tabela 1.

Tema	Título	Autor
Artigos relacionados ao modelo de SGBDs NoSQL	<i>A comparison between several NoSQL databases with comments and notes</i>	[Tudorica e Bucur 2011]
	<i>A Survey of Large Scale Data Management Approaches in Cloud Environments</i>	[Sakr <i>et al.</i> 2011]
	<i>A Request Skew Aware Heterogeneous Distributed Storage System Based on Cassandra</i>	[Zhen e Shanping 2011]
	<i>Implementation of NoSQL for robotics</i>	[Vijaykumar 2010]
	<i>Will NoSQL Databases Live Up to Their Promise?</i>	[Leavitt 2010]
	<i>Cache and consistency in NOSQL</i>	[Peng, Ruichun e Zhiming 2010]
Artigos sobre critérios para seleção e avaliação de SGBD	<i>Evaluating database management systems</i>	[Davidson 1982]
	<i>Data Quality in Context;</i>	[Strong, Lee e Wang 1997]
	<i>Methodologies for data quality assessment and improvement</i>	[Batini <i>et al.</i> 2009]
	<i>Selection of Software: The Analytic Hierarchy Process</i>	[Min 1992]
	<i>Parallel database systems: The future of high performance database systems</i>	[Dewitt e Gray 1992]
	<i>Evolutionary query for heterogenous distributed database systems</i>	[Ghaemi, Tabatabaee e Sadeghizadeh 2008]

**Tabela 1. Artigos selecionados na revisão sistemática**  
Fonte: Próprio autor



## 4. Apresentação dos Resultados

Os critérios para seleção e avaliação de SGBDs foram extraídos da revisão sistemática e sua fundamentação teórica foi complementada pelo trabalho de Elmasri e Navathe (2005). Esses critérios estão reunidos em três grupos e estão apresentados a seguir.

### 4.1. Fornecedor

Após a compilação da lista de SGBDs candidatos, cada fornecedor de software deve ser solicitado para enviar a sua proposta, que pode dar uma visão geral dos recursos que o pacote SGBD oferece [Davidson 1982]. Os fornecedores devem ser solicitados a enviar uma cópia do último relatório anual, no caso de empresas de capital aberto, ou uma descrição da atuação e das atividades da empresa com informações financeiras para avaliar o crescimento e a estabilidade do fornecedor. Os critérios listados a seguir são usados para avaliar os SGBDs em relação ao fornecedor.

**a) Reputação.** Segundo Davidson (1982) uma boa reputação está relacionada às respostas obtidas com os pedidos de informação, à forma como foram tratados os problemas relatados e ao apoio técnico fornecido. Para Min (1992), a reputação do fornecedor pode ser considerada como o seu círculo de negócio, o que pode ser uma boa referência para avaliação da credibilidade, boa vontade e confiança no fornecedor.

**b) Suporte.** Representa o apoio dado pelo fornecedor para solucionar problemas com a operação do produto. Alguns desses serviços são gratuitos, outros funcionam vinte e quatro horas por dia incluindo fins de semana.

**c) Recursos adicionais.** Entre os recursos adicionais, três se destacam: (1) fornecimento de bases de testes para avaliação do produto; (2) treinamento do cliente quanto ao uso do SGBD; e (3) disponibilidade de serviço de pós venda [Davidson 1982].

**d) Continuidade.** Deve-se verificar o compromisso do fornecedor em manter o produto atualizado e de acordo com o cenário de constante evolução da TI. Alguns SGBDs possuem a tendência de estabilizar conforme amadurecem e podem não ser capazes de suprir as necessidades futuras. As melhorias e modificações são importantes para que o SGBD possa acompanhar a evolução das aplicações [Davidson 1982]. Segundo Min (1992), é desejável escolher um fornecedor estabelecido que esteja em atuação por um longo tempo.

### 4.2. Produto

Os requisitos operacionais e funcionais que o SGBD deve satisfazer precisam ser documentados. Os requisitos são utilizados para documentar, categorizar e analisar cada pacote candidato. Quando os requisitos priorizados tiverem sido documentados, eles podem ser usados para desenvolver uma lista de critérios para avaliação de SGBDs candidatos. A seguir destacam-se critérios de avaliação que devem ser considerados em relação ao produto.

**a) Desempenho.** Critérios relevantes para análise de características de desempenho de um SGBD são: otimização de consultas, execução nativa ou em modo de compatibilidade, capacidade de acessar múltiplas bases e paralelismo.

**b) Administração.** Recursos de administração auxiliam os profissionais no gerenciamento e modelagem de dados a realizarem alterações. É desejável que um SGBD forneça bons recursos a fim agilizar alterações no modelo de dados.

**c) Disponibilidade.** Precisa ser avaliado se o SGBD deve ser desativado para execução de funções administrativas. Outro aspecto importante é o processamento concorrente, pois uma aplicação pode exigir a capacidade de suportar vários usuários simultâneos.

**d) Flexibilidade.** O SGBD deve possuir dicionário de dados integrado com linguagem de consulta e disponibilizar geração dinâmica ou supressão de chaves do usuário e privilégios de acesso de forma flexível. Além disso, deve suportar múltiplos usuários e múltiplos bancos de dados, assim como processamento distribuído.

**e) Licença de uso.** Pode ser proprietária ou aberta. Na licença proprietária a distribuição do SGBD é controlada e há custos inerentes a contratação do software. Na licença aberta o SGBD pode ser vendido ou disponibilizado gratuitamente.

### 4.3. Organização

O ambiente de atuação da organização, sua disposição para adotar inovações e assumir riscos podem influenciar os critérios para avaliação e seleção de SGBDs. Segundo Davidson (1982), vários critérios econômicos e organizacionais afetam a escolha de um SGBD, como os descritos a seguir.

**a) Custo.** Alguns tipos de custos devem ser considerados no processo de avaliação. Entre eles se destacam: custo de aquisição do SGBD, manutenção, aquisição de hardware, criação e conversão do banco de dados, pessoal, treinamento e operacional. Além disso, há custos indiretos que devem ser considerados.

**b) Inovação.** A inovação faz parte da filosofia da organização que pode valorizar novas tecnologias ou apenas priorizar a continuidade das aplicações privilegiando a robustez e a conformidade. Este critério pode ser determinante para a avaliação do uso de um SGBD NoSQL, pois este é um modelo inovador.

**c) Perfil de risco.** Este critério considera se a organização está disposta a investir em novas tecnologias, mesmo diante dos riscos e dificuldades encontrados durante processos de mudança.

**d) Importância da TI.** Está relacionado à importância que a organização atribui a TI como fator de adição de valor ao negócio.

**e) Profissional de TI.** Devem-se considerar também os profissionais da organização envolvidos no processo de avaliação e seleção de um SGBD. Eles representam atores interessados no processo de seleção e podem ter comportamentos diversos.

### 4.4. Qualidade

A qualidade de um pacote de software deve ser avaliada usando um modelo de qualidade definido [Koscianski *et al.* 1999]. O modelo de qualidade apresentado neste trabalho com o objetivo de identificar critérios de seleção de SGBDs foi o SQuaRE que faz parte da série ISO/IEC 9126 que categoriza os atributos de qualidade de software em



seis características, as quais são subdivididas em subcaracterísticas que podem ser medidas por meio de métricas internas e externas.

**a) Funcionalidade.** Pode-se afirmar que esta característica de funcionalidade é idêntica aos “requisitos funcionais”, ou seja, os serviços que se espera que o sistema apresente. Ela pode ser decomposta nas seguintes subcaracterísticas: adequabilidade, acurácia, nível de segurança e *multi-threading*.

**b) Manutenibilidade.** Está relacionada ao esforço necessário para fazer modificações específicas no software. Basicamente envolve a facilidade de modificação de um produto de software. A manutenibilidade pode ser decomposta nas seguintes subcaracterísticas: modificabilidade, analisabilidade, estabilidade do produto e testabilidade.

**c) Usabilidade.** Representa o conjunto de atributos que evidenciam o esforço necessário para se utilizar o software, bem como o julgamento individual desse uso. A usabilidade pode ser decomposta nas seguintes subcaracterísticas: operabilidade, compreensibilidade, apreensibilidade e atratividade.

**d) Confiabilidade.** Está relacionada com um conjunto de atributos que evidenciam a capacidade de manter certo nível de desempenho sob determinadas condições durante um período de tempo estabelecido. A confiabilidade pode ser decomposta nas seguintes subcaracterísticas: tolerância e recuperabilidade.

**e) Eficiência.** Trata-se do conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas. A eficiência pode ser decomposta nas seguintes subcaracterísticas: velocidade de operação de um software e utilização de recursos.

**f) Portabilidade.** Trata-se do conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro, ou seja, de ser utilizado em diferentes plataformas. Pode ser decomposta nas seguintes subcaracterísticas: instabilidade, adaptabilidade, coexistência e substitutibilidade.

## 5. Avaliação dos Resultados

Os resultados foram analisados junto a um grupo de profissionais que atuam ou têm experiência em SGBD NoSQL, tanto em atividades profissionais do mercado de TI quanto em atividades acadêmicas. Para seleção desse grupo de profissionais foram considerados os seguintes requisitos: (1) anos de experiência de TI; (2) grau de conhecimento e vivência em projetos com banco de dados; (3) participação em processo de seleção de SGBDs NoSQL; (4) formação acadêmica na área de TI; e (5) domínio no desenvolvimento de aplicações ou de projetos suportados por SGBDs NoSQL.

Os profissionais selecionados foram agrupados em três perfis: (1) DAs e DBAs, que representam os profissionais que atuam diretamente com SGBDs NoSQL, compondo o grupo de administração de dados; (2) Desenvolvedores, que representam os profissionais com a visão de como os SGBDs NoSQL impactam no projeto e desenvolvimento de aplicações; (3) Demais Profissionais de TI, composto por gerentes, fornecedores de SGBDs e acadêmicos.

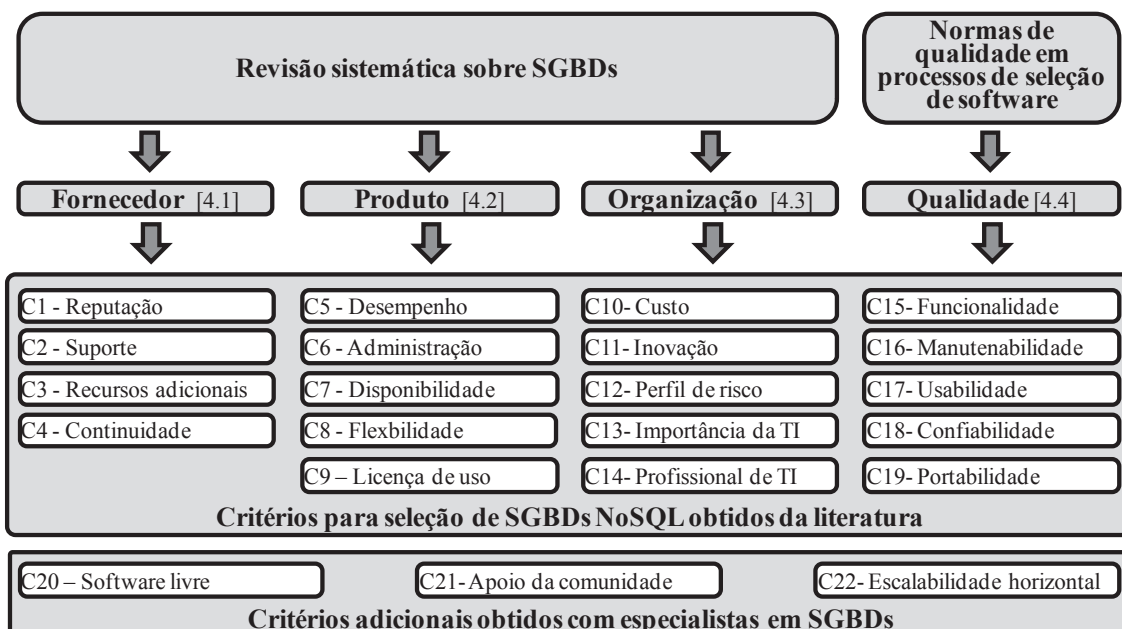
Cada perfil tem visões diferentes sobre o processo de seleção de SGBDs, o que permitiu ao grupo ter uma abordagem mais ampla sobre o tema, mesclando experiências

e conhecimentos práticos dentro do contexto organizacional. Isso permitiu uma abordagem técnica e gerencial sobre os critérios para seleção de SGBDs NoSQL. A Tabela 2 mostra o resumo do perfil do grupo de 32 profissionais que participaram da avaliação dos critérios de seleção de SGBDs NoSQL.

Grupo de Painelistas	Experiência TI (anos)		Experiência SGBD		Part. Seleção SGBD		Formação Acadêmica	
	Até 10	Acima de 10	Baixa	Alta	Não	Sim	Até graduação	Pós-graduação
	(18)	(14)	(1)	(31)	(18)	(14)	(28)	(4)
Desenvolvedores	91,7	8,3	0,0	100,0	91,7	8,3	100,0	0,0
DAs e DBAs	25,0	75,0	0,0	100,0	8,3	91,7	100,0	0,0
Demais Profissionais de TI	50,0	50,0	12,5	87,5	75,0	25,0	50,0	50,0

**Tabela 2. Características do Grupo de Painelistas**  
 Fonte: Próprio autor

O instrumento utilizado para avaliação dos especialistas foi um questionário on-line. A aplicação do instrumento pela internet foi feita entre os meses de janeiro e março de 2013, utilizando a ferramenta Survey Monkey©. Ao responder o questionário, cada especialista confirmou, ou não, o grau de importância dos critérios de seleção de SGBD NoSQL identificados na literatura. Todos os critérios identificados na literatura foram avaliados como importantes pelos especialistas, e eles ainda acrescentaram mais três critérios, conforme indicado na Figura 5: Software Livre, Apoio da comunidade e Escalabilidade horizontal.



**Figura 5. Critérios para Seleção de SGBD**

Seguem as descrições dos três critérios incluídos pelos especialistas:

**a) Escalabilidade horizontal.** Trata-se em escalar as aplicações com o aumento na capacidade de processamento com o acréscimo no número de máquinas no *cluster* utilizando de processamento distribuído.

**b) Apoio da comunidade.** O critério Apoio da comunidade verifica a facilidade em se encontrar informações e experiências sobre determinado SGBD nas comunidades de pesquisa e fóruns na *web*.

**c) Software livre.** Este critério avalia a importância sobre a distribuição gratuita do SGBD NoSQL.

## 6. Considerações Finais

A cada dia, e com maior intensidade nos últimos anos, o tema do *Big Data* cresceu em importante no contexto das aplicações web, no qual o crescente volume, variedade e velocidade de informações tem-se tornado uma preocupação para as organizações. Nesse contexto, surgem os SGBDs NoSQL com o objetivo de prover uma modelagem de dados simplificada, possibilitando ganhos de desempenho, maior grau de escalabilidade, flexibilidade e disponibilidade. Por outro lado, os SGBDs NoSQL não são uma novidade simples de ser adotada. Por essa razão, identificar critérios para seleção de SGBDs NoSQL se mostra importante.

Esta pesquisa contribuiu para identificar os critérios mais importantes para seleção de SGBDs NoSQL. Ela identificou na literatura os principais critérios de seleção de SGBDs NoSQL e os verificou junto a um grupo de 32 especialistas no tema. Como resultado, a pesquisa identificou 22 critérios que devem ser observados pelas organizações em seus processos de seleção de SGBDs NoSQL.

Como próxima etapa de pesquisa, pretende-se analisar o grau de importância desses 22 critérios de modo a se obter não apenas um *ranking* desses critérios, mas identificar aqueles que são fundamentais no processo de seleção.

## Referências

- Batini, C., Cappiello, C., Francalanci, C. e Maurino, A. (2009) “Methodologies for data quality assessment and improvement”, *ACM Computing Surveys*, 41(3), p. 1-52.
- Chang, F., Dean, J. e Ghemawat, S. B. (2008) “A distributed storage system for structured data”, *ACM Transactions on Computer Systems (TOCS)*, 26(2), article n. 4.
- Davidson, E. (1982) *Evaluating database management systems*. In *Proceedings of the June 7-10, national computer conference (AFIPS '82)*. ACM, New York, NY, USA.
- Dean, J. e Ghemawat, S. (2008) “MapReduce: simplified data processing on large clusters”, *Communications of the ACM*, p. 1-1.
- Dewitt, D. e Gray, J. (1992) “Parallel database systems: The future of high performance database systems”, *Communication ACM* 35.
- Elmasri, R. e Navathe, S. B. (2005) “Sistemas de Banco de Dados”, São Paulo: Pearson Education do Brasil.

- Ghaemi, R., Fard, A. M., Tabatabaee, H. e Sadeghizadeh, M. (2008) “Evolutionary query for heterogeneous distributed database systems”, World Academy of Science, Engineering and Technology, p. 43-49.
- Ghemawat, S., Gobioff, H. e Leung, S. (2003) “The Google file system”, SIGOPS Operating Systems Review, 37, p. 29-43.
- Leavitt, N. (2010) “Will NoSQL Databases Live Up to Their Promise?”, IEEE Computer, 43(2), p. 12-14.
- Lóscio, B. F., Oliveira, H. R. e Pontes, J. C. S. (2011) “NoSQL no desenvolvimento de aplicações Web colaborativas”, VIII Simpósio Brasileiro de Sistemas Colaborativos.
- Min, H. (1992) “Selection of Software: The Analytic Hierarchy Process” International Journal of Physical Distribution & Logistics Management, p. 42-52.
- Pokorny, J. (2011) NoSQL databases. A step to database scalability in web environment. In Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services (iiWAS '11). ACM, New York, NY, USA, 278-283.
- Rahien, A. (2010) “That No SQL Thing: Column (Family) Database”, Disponível em: <<http://ayende.com/blog/4500/that-no-sql-thing-column-family-databases>>. Acesso em 14/01/2013.
- Sadalage, P. J., Fowler, M. e River, U. S. (2013) “NoSQL distilled: a brief guide to the emerging world of polyglot persistence”, Pearson ed.
- Sakr, S., Liu, A., Batista, D. M. e Alomari, M. (2011) “A Survey of Large Scale Data Management Approaches in Cloud Environments”, IEEE Communications Surveys & Tutorials, 13(3), p. 311-336.
- Strong, D. M., Lee, Y. W. e Wang, R. Y. (1997) “Data Quality in Context”, Communication ACM 40, p. 103-110.
- Tiwari, S. (2011) “Professional NoSQL”, Indianapolis: John Wiley & Sons, Inc.
- Tudorica, B. G. e Bucur, C. (2011) “A comparison between several NoSQL databases with comments and notes”, 10<sup>th</sup> Roedunet International Conference, p. 1,5, 23-25.
- Vijaykumar, S. (2010) “Implementation of NOSQL for robotics”, Emerging Trends in Robotics and Communication Technologies (INTERACT), p. 195-200.
- Xiang, P., Hou, R. e Zhou, Z. (2010) “Cache and consistency in NOSQL”, 3rd International Conference on Computer Science and Information Technology, p. 117-120.
- Ye, Z. e Li, S. (2011) “A request skew aware heterogeneous distributed storage system based on Cassandra”, Computer and Management (CAMAN), p. 19-21.