

Programação Orientada e Objetos II



Anhanguera

AVALIE
SUA PROFISSÃO

QUANDO APARECER EM SEU
PORTAL UMA AVALIAÇÃO SOBRE
SEU CURSO, RESPONDA:



NOTAS

9 ou 10

SIGNIFICA QUE VOCÊ INDICA

NOTAS

7 ou 8

SIGNIFICA QUE VOCÊ NÃO INDICA



Anhanguera



Anhanguera



A computação é uma ciência nova no mundo comparado à física ou matemática. As formas de armazenamento, processamento e transmissão dos dados muda e avança a cada geração de computadores, softwares e métodos. Existem diversos marcos que culminaram em padrões que são utilizados até hoje, a própria arquitetura e organização dos computadores são referências a projetos antigos, e esse processo também ocorreu com os sistemas de gerenciamento de banco de dados (SGBD). Antes de existir uma forma padronizada de armazenamento, consulta, alteração e remoção, os dados eram armazenados em arquivos e as aplicações faziam acesso direto a eles, assim, as formas de acesso e controle de concorrência eram feitas diretamente pela aplicação. Nesse cenário, qualquer defeito poderia causar inconsistências e problemas com perda de informação (PLUGGE; MEMBREY e HAWKINS, 2010). Além disso, cada programador precisa aprender uma forma de fazer acesso aos dados, o que leva a aplicações mais especializadas e mais dependentes de profissionais específicos.



Anhanguera

Em 1970 foram criadas as primeiras versões dos SGDB relacional (tabelas e suas relações com chaves) e da linguagem de consulta estruturada (SQL, do inglês Structured Query Language), iniciando o padrão de armazenamento de informação que vigora até hoje para as mais diversas aplicações (HEUSER, 2011). Todavia, com os avanços no hardware e aumento exponencial na computação, contando com celulares, notebooks, smartphones e diversos outros dispositivos que fazem parte do cotidiano, surgiu a necessidade de outra forma de controle dos dados. Hoje, existe muita informação disponível e com necessidade de ser processada, chegando a ser chamada atualmente de big data (HAN, E e LE, 2011).



Anhanguera

Os métodos para tratar grande volume de informação têm como foco acessar e tratar relacionamentos não triviais nos dados, ou ainda, realmente entender o que está armazenado, pois os dados podem não estar em tabelas ou serem relacionados. Com isso, temos cenários de aumento constante no volume de informação e com um agravante sério a serem tratados; em diversos casos, esses dados não são estruturados (não possuem uma forma organizada de recuperar a informação) e ainda não são nem textuais, nem numéricos, são vídeos ou imagens. Além disso, caso exista uma estrutura, ela pode ser extremamente dinâmica, causando dificuldade em sua modelagem.



Anhanguera

A Figura 4.1 apresenta uma visão, com exemplos, de como é a quantidade de informação relacionando a aplicação e sua complexidade de variedade. Olhando o primeiro quadrante da Figura (o quadro menor) temos o planejamento de recursos da empresa (EPR, do inglês Enterprise Resource Manager) contendo, em sua maioria, informações estruturadas e alocadas em tabelas com seus tipos de dados bem definidos e ainda todos os dados relacionados. Esse tipo de cenário consiste em um sistema no qual a entrada de dados é controlada de maneira a combinar com os tipos da tabela, sem liberdade ao usuário. A quantidade de dados em um sistema ERP é relacionada diretamente ao número de operações, todavia, como é textual, não ocupa tanto espaço como um conjunto de imagens que são aplicados no próximo cenário. No segundo quadrante, podemos alocar os dados de gestão de relacionamento com o cliente (CRM, do inglês Customer Relationship Manager), em que se utiliza a base do ERP, completada com outros dados, considerando diversos conjuntos de propagandas, documentos com folhetos, imagens e outros.

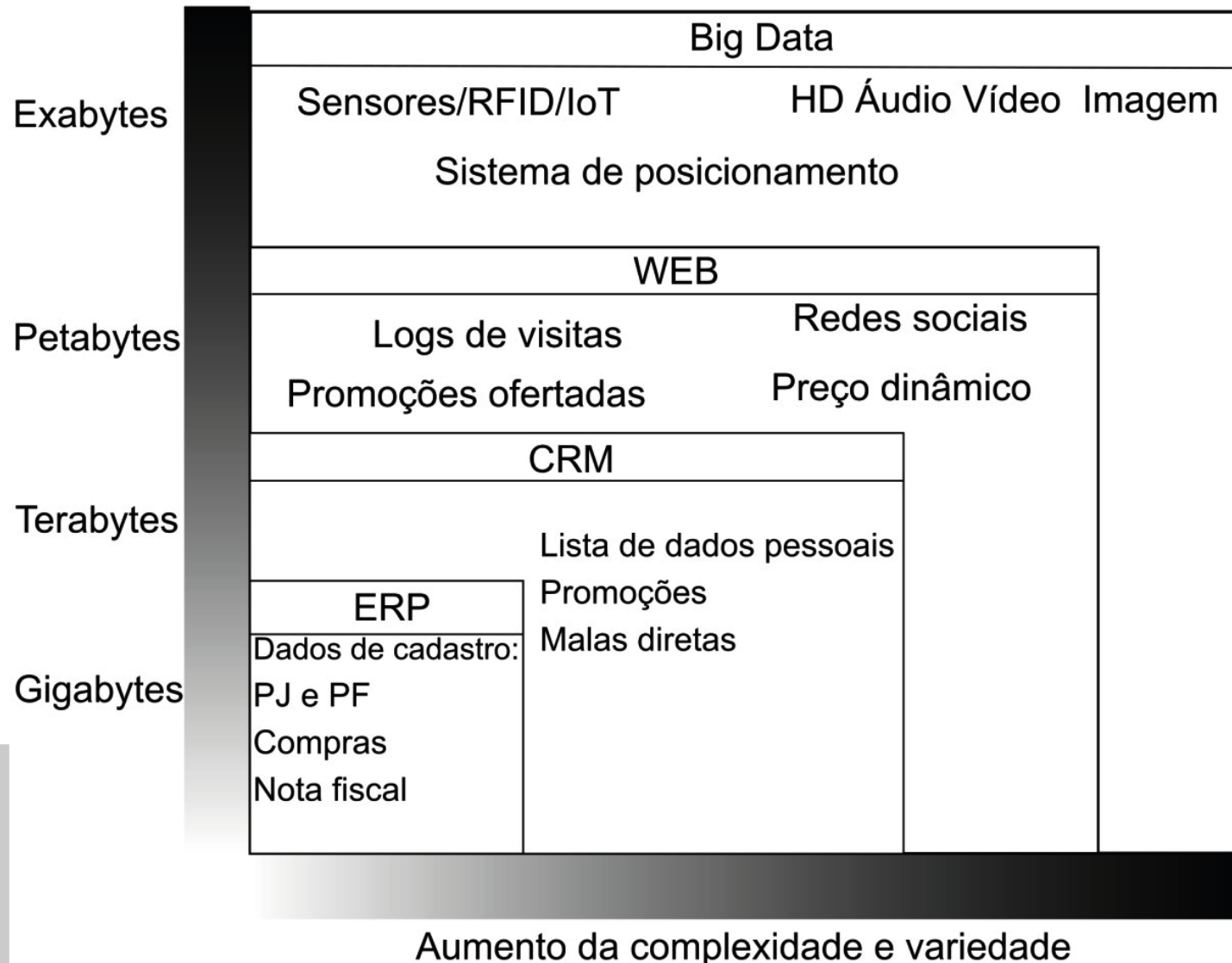


Anhanguera

Ainda no CRM, mesmo tendo mais dados que o ERP, os dados também são estruturados, sendo alocados de forma organizada pela interface com valores controlados e legíveis pelo computador. Ao avançar para o próximo quadrante (Web), já começamos a considerar cenários mais dinâmicos, nos quais o volume de informação cresce de forma muito mais rápida. Sendo aqui considerados imagens, vídeos, áudios e outros conjuntos de informação que já não são mais legíveis pelo computador, necessitando de uma análise para se entender sua semântica. Unindo todos esses elementos com as informações de sensores, identificação por rádio frequência (RFID, do inglês radio frequency identification) e internet das coisas (IoT, do inglês Internet of Things), chegamos a um cenário onde temos muitas informações diversificadas. Com isso, a semântica necessária é aplicada por processos de inteligência artificial, aprendizado de máquina, séries temporais (time series) e outras formas de entendimento por máquina.



Anhanguera





Unindo os argumentos já citados com outros itens, podemos resumir os diversos desafios de armazenamento de dados do período atual na lista:

1. Grande volume de informação;
2. Informações de diversos tipos, em forma estruturada e não estruturada;
3. A onipresença do software e a constância de mudanças;
4. Métodos ágeis de desenvolvimento, nos quais as mudanças de uma aplicação são bem-vindas;
5. Cenários em que a redução de custos é sempre um objetivo das empresas.



Dados todos os argumentos e cenários, podemos considerar que até hoje, os SGDB relacionais são suficientes para resolver esse tipo de problema. Todavia, existem diversos custos para que esse tipo de banco de dados cumpra todos os novos requisitos, como exemplo:

1. Custos de hardware para escalabilidade de volume de dados e desempenho;
2. A construção das tabelas de um SGDB relacional e suas consultas é ligada a aplicação;
3. Demanda de pessoal especializado para manter o desempenho do SGBD;
4. Em sistema orientado a objetos, é necessário criar tanto a modelagem das classes para serem utilizadas no sistema quanto o modelo de armazenamento.



Anhanguera

Para atender essa demanda latente do novo cenário de volume de informação, em 2006 aconteceram diversos esforços vindos de empresas como Google e Amazon (LEAVITT, 2010). Tais empresas possuem grande quantidade de dados a serem tratados e, em diversos casos, essas informações necessitam ser divididas em vários computadores. Frente a esses movimentos foram criados/revisitados os conceitos de NoSQL. Esse “novo” modelo de armazenamento de informação foi proposto a partir da necessidade de sistemas de banco de dados relacionais mais leves, já sendo trabalhado desde 1998. O termo NoSQL se refere a “Not Only SQL”, ou seja, não apenas SQL (PLUGGE, MEMBREY e HAWKINS, 2010). O nome faz analogia a não obrigatoriedade de usar uma linguagem de consulta estruturada (por exemplo, o SQL), pois o próprio modelo de armazenamento não é estruturado na forma de tabelas e colunas.



Anhanguera

De forma direta, os bancos de dados NoSQL tratam a informação como um formato em que a estrutura não precisa ser definida. Ainda, a forma de buscar a informação não obriga o programador a saber como os dados estão organizados, diferente do modelo relacional. Dessa forma, o desenvolvimento se torna mais dinâmico e mais adaptável para as constantes mudanças que podem ocorrer durante o processo, deixando assim mais próximo o desenvolvimento das metodologias ágeis.



Anhanguera

Porém, antes de analisarmos os conceitos de um banco de dados NoSQL, é importante deixar claro que ele não é a solução para todos os casos. Cenários mais clássicos, como instituições financeiras, sistemas legados e operações críticas em que a relação de diversas tabelas é necessária, o uso do NoSQL deve ser analisado com cautela. Uma das limitações dos sistemas NoSQL refere-se a propriedade de atomicidade das operações, lembrando que a atomicidade é uma característica do SGBD, na qual uma operação deve ser executada por completo ou falhar, por exemplo, a transferência de dinheiro entre contas bancárias. Com isso, a utilização de NoSQL em cenários em que esse tipo de operação é essencial deve ser analisada com cuidado.



O NoSQL possui um conjunto de características básicas que devem ser utilizadas para guiar a tomada de decisão no uso dessa nova tecnologia, tais como as descritas por HAN, E e LE (2011):

1. Independente de esquema: o armazenamento não possui uma organização rígida, levando a um conjunto de bibliotecas para acesso mais simples;
2. Livre de junções de tabelas: não são indicados os joins, mesmo que possíveis em alguns modelos, tornando a utilização menos complexa;
3. Crescimento horizontal: a maioria das soluções NoSQL são escaláveis, sendo anexado mais computadores e não necessariamente computadores mais caros;
4. Hardware mais simples: é possível utilizar mais hardware barato pela própria arquitetura da solução;
5. Autocontido: cada computador que funciona como armazenamento é independente, levando em diversos casos cópias das bases de dados;
6. Controle de usuário: necessita de poucos elementos de controle de usuário, transação e concorrência.



Modelos de dados para os NoSQL

Como a estrutura de acesso a informação em um NoSQL é simples, torna-se possível variar a estrutura interna de controle. Existem muitos modelos para organizar a informação dentro de um NoSQL, todavia, existem quatro modelos mais comuns (GESSERT et al., 2016):

1. Key-value;
2. Document stores;
3. Orientado a colunas;
4. Grafo.



Como o NoSQL é uma tecnologia em crescimento e que vem conquistando seu espaço no mercado, é complicado afirmar qual é o melhor modelo. Existem muitas variáveis a serem consideradas, e cada projeto de NoSQL afirma que as suas formas de implementação são as melhores. Assim, para cada cenário é importante avaliar onde cada solução foi efetivada e seus resultados.

O sistema de key-value é uma estrutura mais simples, em que cada conjunto de dados possui apenas uma chave principal. O Quadro 4.1 apresenta um exemplo deste sistema, onde na coluna Key temos um número que identifica o registro e na coluna value existem os valores armazenados. Repare que não é necessário padronizar os dados, sendo que os valores da entrada 1 são diferentes da entrada 2.



Anhanguera

Dados	
Key	Value
1	Aluno: João Curso: Ciência da Computação Conclusão: 2020
2	Aluno: Pedro Curso: Ciência da Computação Conclusão: 2022 Origem: Transferência interna



Anhanguera

Um exemplo desse tipo de modelo é o Amazon SimpleDB. Esse tipo de organização permite as seguintes vantagens:

1. Flexibilidade na modelagem: os Values podem ser inseridos e alterados relacionado a cada registro;
2. Operação simples: é necessário apenas buscar a Key e já se tem acesso aos dados;
3. Escalabilidade e disponibilidade: pela sua simples estrutura, as técnicas computacionais para aumentar a capacidade e redundância são facilmente aplicadas.



Outro modelo muito utilizado é o document stores, em que se utiliza diversos arquivos formatados para fazer o armazenamento dos dados. Nesse modelo, em diversos casos, se utiliza o Extensible Markup Language (XML) ou o JavaScript Object Notation (JSON) para fazer a formatação dos dados que são armazenados no NoSQL. O Quadro 4.2 apresenta um exemplo de um JSON que pode ser utilizado por um banco de dados NoSQL. Observe que é possível criar diversas estruturas em cada registro, e ainda no próprio NoSQL você pode utilizar diversos arquivos diferentes para guardar os dados.



Anhanguera

```
{  
  '_id' : 1,  
  'name' : { 'Nome' : Pedro, 'Sobrenome' : 'Silva' },  
  'curso' : [ 'Ciência da Computação', 'Análise e  
desenvolvimento de sistemas']  
}
```



Um exemplo de uso desse tipo de implementação é o MongoDB (REDMOND e WILSON, 2012). Esse modelo apresenta as seguintes vantagens:

1. Capacidade organizacional: junto do JSON é possível armazenar arquivos e outros tipos de dados binários;
2. Forma organizada: existem grande quantidade de informação já colocada em JSON, dessa forma a migração é algo menos complicado;
3. Informação variada: pela sua estrutura é possível ter dados em que alguns elementos do registro estão disponíveis e outro não (null).



Anhanguera

O sistema orientado a colunas organiza a informação de maneira semelhante ao modelo de document stores. Todavia, ele cria estruturas separadas para juntar informações pertinentes a uma entidade. Sua aplicação é considerada mais específica, pois os casos de uso são relacionados a sistemas com grande volume de dados já estruturados e com necessidade de processamento muito rápida. O Quadro 4.3 apresenta um exemplo desse sistema; repare que a estrutura dos dados é mais presente.



Família de super coluna: Clientes

RowID: 1

Super Coluna: Nome

Nome: João

Sobrenome: Silva

Super Coluna: Curso

Nome: Ciência da Computação

Conclusão: 2020

RowID: 2

Super Coluna: Nome

Nome: Pedro

Sobrenome: Silva

Super Coluna: Curso

Nome: Engenharia de Computação

Conclusão: 2020



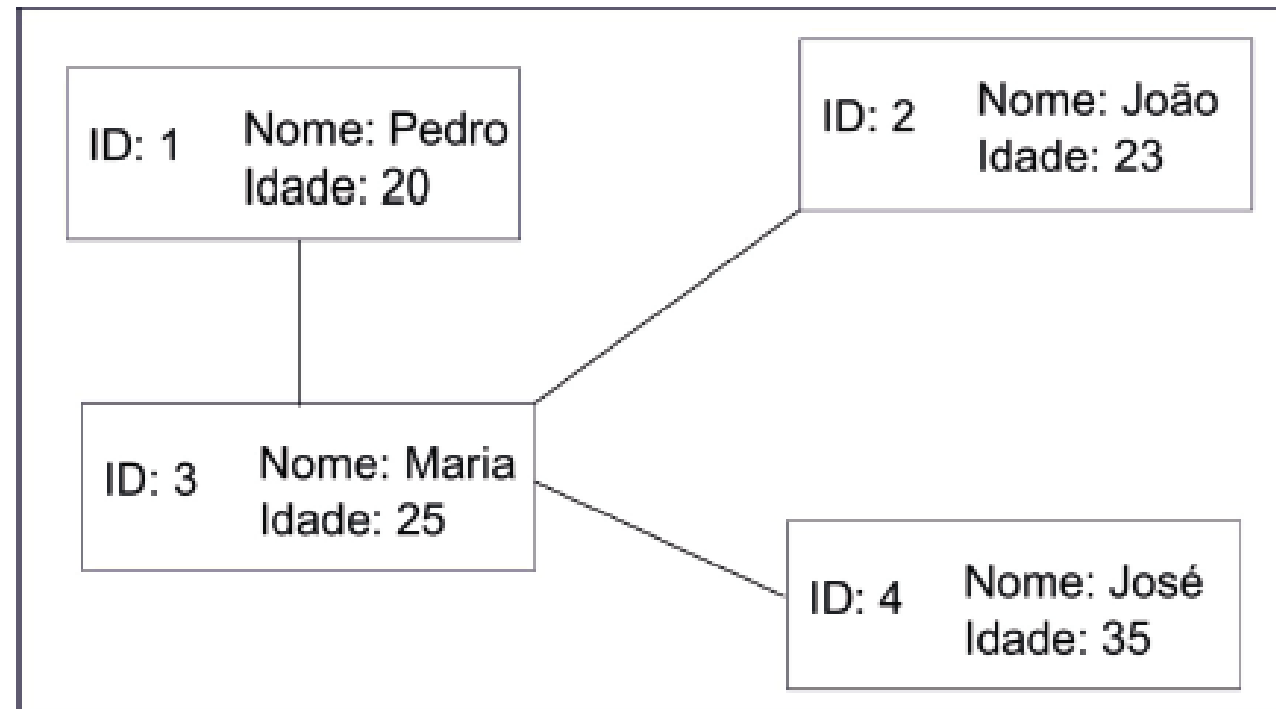
Exemplos de implementação desse modelo é o Apache Cassandra, aplicado para processamento dos dados da Netflix e da Organização Europeia para Pesquisa Nuclear (CERN, do inglês European Organization for Nuclear Research).

O último modelo consiste em uma estrutura de dados baseada em grafos, em que a informação pode ter uma estrutura inicial de key-value. Porém, existe algum tipo de relacionamento entre cada uma das entradas dos dados. A Figura 4.2 apresenta um exemplo desse tipo de forma de armazenamento, no qual o item de ID 1 possui ligação com o item 3, e o item 3 possui ligações com o 2 e 4. Um exemplo desse tipo de modelo é neo4j.



Exemplificando

Figura 4.2 | Sistema de armazenamento por forma de grafos



Fonte: elaborado pelo autor.



Existem diversos cenários em que um NoSQL baseado em grafos pode ser utilizado:

- Sistema de recomendações de filmes;
- Armazenamento de sistema forense;
- Sistema de armazenamento para machine learning;
- Relacionamento entre pessoas em uma rede social;
- Comunicação de sistema de *internet of things* (lot).



Os sistemas de gerenciamento de banco de dados (SGDB) são serviços essenciais para garantir o armazenamento e acesso aos dados. Com os SGDB relacionais é possível compartilhar, proteger um grande conjunto de informação e descrever quais são as ligações dos dados. Todavia, o movimento NoSQL tem o objetivo de tornar esse processo mais simples.:

Qual das características a seguir representa o objetivo de um sistema NoSQL?

- a) Dependência de esquema de estrutura de dados.
- b) Junções de tabelas.
- c) Crescimento vertical.
- d) Independência de esquema de organização.
- e) Dependência de sistema de centralização.



Os modelos NoSQL são relacionados ao tipo de dado que será armazenado, dessa forma, cada projeto deve explicar qual é sua forma interna de armazenamento e explicar aos desenvolvedores quais são os cenários ideais. Dados os diversos cenários que os NoSQL operam, são necessários modelos diferentes para propósitos variados.

Quais são os quatro modelos de NoSQL mais utilizados?

- a) Key-value, document store, orientado a linha e grafo.
- b) Key-key, document store, orientado a coluna e grafo.
- c) Key-table, document store, orientado a coluna e árvore.
- d) Key-value, document store, orientado a coluna e árvore.
- e) Key-value, document store, orientado a coluna e grafo.