

INSTITUTO FEDERAL  
Paraíba  
Campus João Pessoa

Aula

12

# Banco de Dados II

*Atualizada em 02/12/2019*

## Visão Geral do MongoDB



**Professor:**

Dr. Alex Sandro da Cunha Rêgo



[alex@ifpb.edu.br](mailto:alex@ifpb.edu.br)

# Introdução



- Significado
  - Derivado da palavra “*humongous*” em inglês (imenso)
- MongoDB armazena dados em **documentos**
  - “documentos” não são como arquivos Adobe PDF ou Microsoft Word, e sim documentos baseados na especificação **JSON** (*Java Script Object Notation*)
- Um **database** é um container de coleções
  - **Exemplo:** Biblioteca Setorial
- Uma **coleção** é um container para documentos
  - **Exemplo:** livros de informática, administração, engenharia elétrica, construção civil, etc.

# Introdução



- Implementado em C++
  - Facilidade de portar e/ou executar a aplicação praticamente em qualquer lugar
- Distribuição
  - Disponível para as plataformas Linux, Mac OS, Windows e Solaris
    - ✓ Versão de **32 bits**: BD's limitados a até 2GB
    - ✓ Versão de **64 bits**: sem limitação
- JSON ➤ *Schemaless*
  - Dados reunidos em um único local proporcionam um melhor desempenho em operações de consulta/atualização

# Introdução






- **Equivalência**

Conceito Relacional	Equivalente no MongoDB
Database	database
tabelas	<b>coleções</b> (depósito de documentos)
linhas	<b>Documentos</b> (armazena informações complexas tais como <b>listas</b> , <b>dicionários</b> e até mesmo <b>listas de dicionários</b> )
Colunas	par <b>chave-valor</b>
Índice	Índice
<b>Constraints</b> (Chave primária, NOT NULL, etc.)	-----
<b>Transações</b>	-----

# Introdução



- Stack Overflow Survey (2018)

Most Popular Database Platforms  Love  Dread  Want									
	2019	2018	%Change	2019	2018	2019	2018	2019	2018
MySQL	52%	59%	-7%	54%	49%	46%	51%	8%	8%
PostgreSQL	36%	33%	3%	70%	62%	30%	38%	14%	11%
MS SQL Server	34%	42%	-8%	58%	52%	43%	48%	3%	4%
SQLite	30%	20%	10%	56%	48%	45%	52%	7%	53%
MongoDB	26%	26%	0%	60%	55%	41%	45%	18%	45%
Redis	20%	19%	1%	71%	65%	29%	36%	11%	35%
MariaDB	17%	14%	3%	59%	53%	41%	47%	4%	47%
Oracle	16%	11%	5%	38%	37%	62%	63%	3%	63%
Elasticsearch	16%	14%	2%	63%	60%	67%	40%	11%	40%



# Introdução



- Quem utiliza MongoDB?



# Instalação



- **Como instalar o MongoDB manualmente**

- ❑ Acessar o centro de download do MongoDB e selecionar a seção **MongoDB Community Server**
  - ✓ <https://www.mongodb.com/download-center>
- ❑ Selecionar o Sistema Operacional de interesse
- ❑ Extrair o conteúdo da pasta em uma unidade de disco
- ❑ Dentro da pasta extraída (ex: \mongodb\), criar o diretório de dados (ex: c:\mongodb\data)
- ❑ Para realizar o **start** do mongoDb, executar a instrução `.\bin\mongod.exe --dbpath c:\mongodb\data`
  - ✓ Conexões são aguardadas na porta **27017**
- ❑ Conectar ao MongoDB: `shell .\bin\mongo.exe`



Install

# O shell do MongoDB



- Aplicativo **mongo.exe**

```
> version()
4.0.6
> _
```



Se o MongoDB for iniciado com os parâmetros *default* (e o shell também), você iniciará conectado ao *database default* **test**. Este banco é criado automaticamente no instante em que se conectar ao mongo



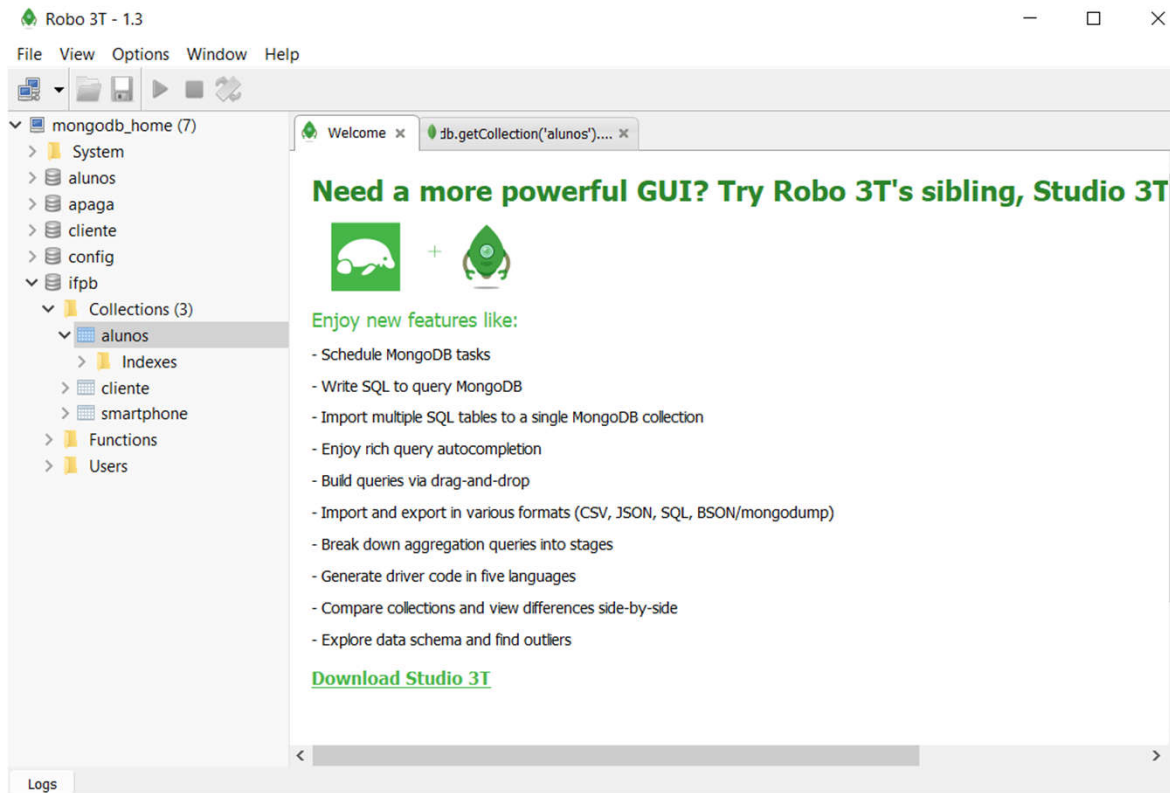
Se tentar se conectar um banco de dados que não existe, o MongoDB criará pra você



# Robot 3T



- Interface Gráfica para o MongoDB



<https://robomongo.org/>

**Robot 3T:** GUI leve e **gratuita** para entusiastas do MongoDB

**Studio 3T:** IDE profissional e completa para o MongoDB. Adiciona diversas funcionalidades (import, export, migration, autocomplete)

# O shell do MongoDB



- **Comandos úteis**

- ❑ `show databases`: exibir os databases disponíveis (ou **dbs**)
- ❑ `use <database>`: define o banco de dados corrente
- ❑ `show <collections>`: exibir as coleções disponíveis (do banco de dados selecionado)
- ❑ `show users` : mostra os usuários conectados ao bd corrente
- ❑ `db.<coleção>.insert()` : inserção
- ❑ `db.<coleção>.update()` : atualização
- ❑ `db.<coleção>.remove()` : remoção
- ❑ `db.<coleção>.find()` : consulta
- ❑ `db.help()`: outras ações aplicadas ao objeto **db**
- ❑ `db.createCollection('name')`: criar uma coleção
- ❑ `db.<collection_name>.drop()`: apagando uma coleção
- ❑ `db.shutdownServer()`: parar o mongoDB
- ❑ `db.dropDatabase()`: apaga o database corrente

# O shell do MongoDB



- **Comandos úteis**

- ❑ `db.version()` : obtém a versão corrente do mongodb
- ❑ `quit()` : sair do console do mongo
- ❑ `cls` : apagar tela
- ❑ `help`: exibe a lista completa de comandos
- ❑ `typeof()`: obtém o tipo de um item de dado
- ❑ `print()`: imprime uma string ou conteúdo de variável na tela

# Mongo Shell

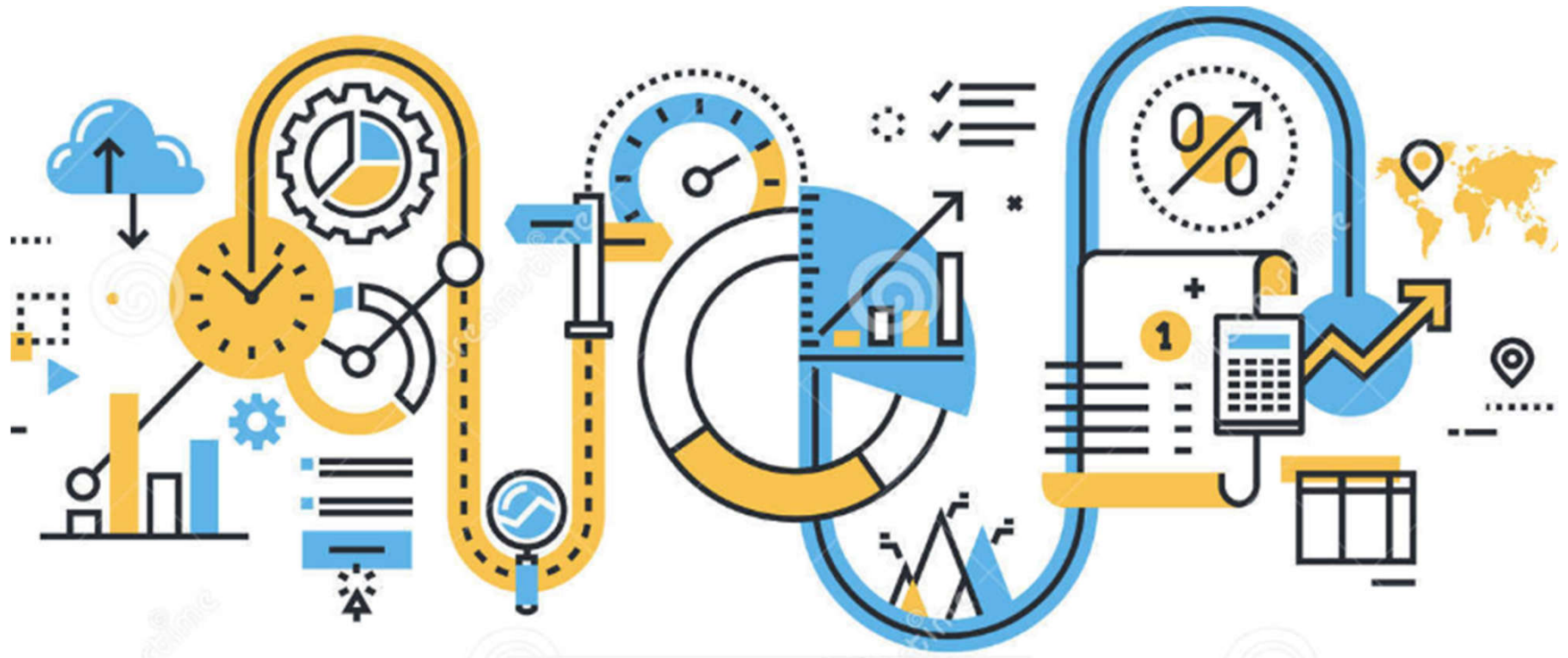


- **Aceita a declaração de variáveis**

```
mongo SHELL
> var d1 = Date()
> var d2 = new Date()
> print('d1 = ', d1, ' / d2 = ', d2)
d1 = Fri Jul 26 2019 13:06:44 GMT-0300 / d2 = Fri Jul 26 2019 13:06:46 GMT-0300
> typeof(d1)
string
> typeof(d2)
object
> d1
Fri Jul 26 2019 13:06:44 GMT-0300
> d2
ISODate("2019-07-26T16:06:46.407Z")
> _
```

- ❑ Objetos ISODate() armazenados em uma variável permitem o acesso às suas propriedades
  - ✓ getHours(), getMinutes(), getSeconds(), getMonth(), getFullYear()

# Modelo de Dados



# Databases



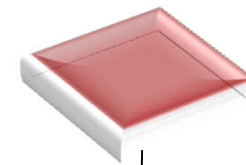
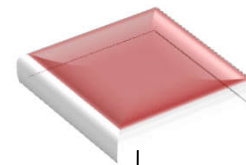
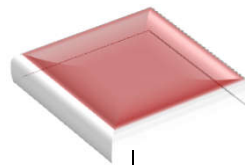
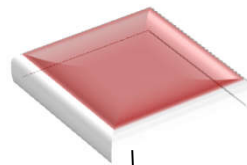
- Repositório central de coleções
  - Conecte a um banco de dados (existente ou não)

```
> use trafego  
> db  
trafego
```



database

coleções



documentos



# Coleções



- Análogas às tabelas, porém menos rígidas
  - ❑ Repositório de itens (documentos) semelhantes
    - ✓ Os documentos não precisam ter estrutura fixa
  - ❑ São criadas por demanda
    - ✓ Basta aparecer a primeira oportunidade de armazenar um documento que a referencie
  - ❑ Toda coleção no MongoDB tem um nome único
    - ✓ Nomes podem conter letras e números
    - ✓ O símbolo \$ é reservado
    - ✓ Número máximo de caracteres: 118 (seja simples...)
  - ❑ Acesso a uma coleção  
**db.<coleção>**
  - ❑ Recebem ações C.R.U.D.

```
> db.infracoes  
trafego.infracoes
```

# Documentos



- Similar à estrutura de um JSON

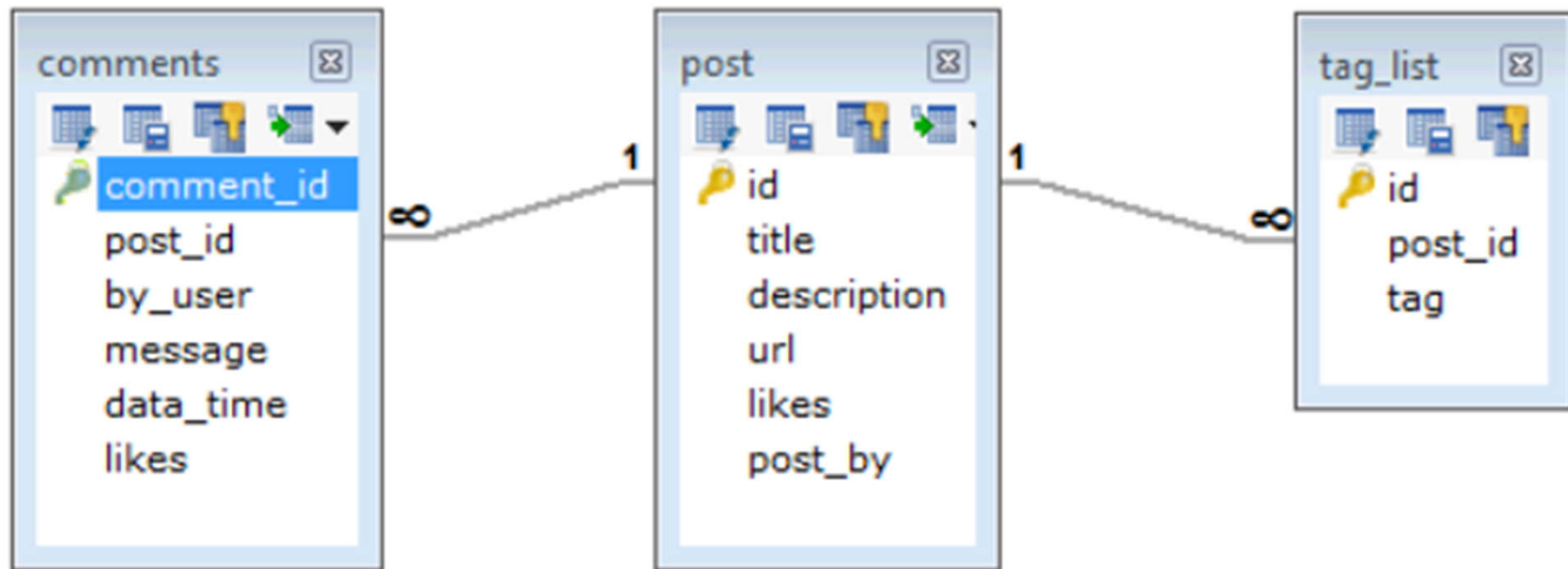
```
{  
  ← chave / valor  
  placa : 'MOV1234',  
  fabricante : 'Honda',  
  modelo : 'Civic',  
  ano: 2019,  
  acessórios: [ ← array  
    'Câmera de ré', 'ar-condicionado digital',  
    'Velocidade de cruzeiro', 'bluetooth'  
  ]  
  ← Documentos embutidos  
  multas: [  
    { data: '2019-02-28', valor: 104.28 },  
    { data: '2018-10-01', valor: 53.35 }  
  ]  
}
```



# Modelagem



- **RDBMS x Orientado a Documentos**



# Modelagem



- **RDBMS x Orientado a Documentos**

```
{
  _id: POST_ID
  title: TITLE_OF_POST,
  description: POST_DESCRIPTION,
  by: POST_BY,
  url: URL_OF_POST,
  tags: [TAG1, TAG2, TAG3],
  likes: TOTAL_LIKES,
  comments: [
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    },
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    }
  ]
}
```

# Formato de Armazenamento



- **BSON (Binary JSON)**

- ❑ padrão aberto de dados, usado pelo MongoDB, que engloba não só a parte textual típica de um JSON (pares chave-valor), como também o armazenamento de documentos em formato real
  - ✓ Elimina a necessidade de um mapeador ou código especial para fazer a **conversão** de um documento
- ❑ O BSON foi pensado para prover maior facilidade de ser percorrido e indexado
  - ✓ Mesmo que ocupe mais espaço em disco do que o JSON
- ❑ O tipo BSON permite que o dado seja facilmente convertido para um formato de dados nativo de uma linguagem de programação
  - ✓ Uso de drivers para Python, Ruby, PHP, C, C++ e C#

# Formato de Armazenamento



- **BSON (Binary JSON)**

- ❑ Suporta o armazenamento de até 4MB de dados binários
- ❑ Não está limitado ao armazenamento, apenas, de dados textuais (PDF, Word, txt, etc.)
  - ✓ Pode armazenar imagens, clipes de áudio, etc.



Explorar a especificação **GridFS**

# Tipos de Dados BSON



- **Tipos fundamentais**

- ❑ **String:** representação de dados textuais (entre “” ou “”)

```
{ placa : 'MOV1234' }
```

- ❑ **Integer:** armazenar valores numéricos

```
{ ano : 2019 }
```

- ❑ **Booleano:** valores definidos como TRUE ou FALSE

```
{ casado : TRUE }
```

- ❑ **Double:** armazenamento de valores reais

```
{ salario : 1025.28 }
```

- ❑ **Arrays:** lista de dados

```
{ irmãos : ['Joao', 'Ana', 'Letícia'] }
```

# Tipos de Dados BSON



- **Tipos fundamentais**

- **Timestamp**: representação de um momento

```
{ ts : new Timestamp() }
```

- **Date**: armazenar data ou hora corrente (ISO Date)

```
{ data1: new Date(2019,02,28,16,55),  
  data2: Date()} //retorna data atual como string
```

- **ISODate**: armazenar data no format ISODate

```
{ data : ISODate('2012-12-19T06:01:17.171Z') }
```

- **ObjectID**: armazenar ID do documento

```
{ _id: ObjectID('542c2b97bac0595474108b48') }
```

- **Null**: armazenar um valor 'nulo'

```
{ endereço: Null }
```

# Tipos de Dados



- **GridFS**

- ❑ Solução MongoDB para o armazenamento de dados acima de 4 MB
- ❑ O uso do GridFS é possível por meio do driver MongoDB para a linguagem de programação
- ❑ Os metadados sobre o arquivo são armazenados na coleção **fs.files**, enquanto os dados propriamente ditos são separados em partes chamadas **chunks** (porções), as quais são armazenadas na coleção **fs.chunks**

[https://www.tutorialspoint.com/mongodb/mongodb\\_gridfs](https://www.tutorialspoint.com/mongodb/mongodb_gridfs)

# Referências Bibliográficas



- MongoDB Tutorial (Quackit).  
<https://www.quackit.com/mongodb/tutorial/>
- MongoDB Tutorial (Tutorialspoint):  
<https://www.tutorialspoint.com/mongodb/index.htm>
- Tutorial MongoDB para Iniciantes em No-SQL:  
<https://www.luiztools.com.br/post/tutorial-mongodb-para-iniciantes-em-nosql-2/>
- Hows, D.; Membrey, P., Plugge, E.  
**Introdução ao MongoDB.** São Paulo: Novatec, 2015.