

# Databases noSQL

- **Prof. Vinicius Aquino do Vale** (<https://www.linkedin.com/in/aquinovale/>)
- MBA em Big Data.
- Bacharelado em Ciência da Computação.
- Curso Técnico em Computação pelo ETEIT
- Curso de Informática Aplicada pelo SENAI
- Desde Mar/2016 atuando como Big Data Engineer
- Atuou como consultor sênior responsável pela administração da infraestrutura e arquitetura de ambientes na Caixa Econômica Federal.
- 10 anos de experiência em administração de sistemas operacionais Linux (Red Hat) para ambientes transacionais (PostgreSQL/Big Data).
- 4 anos de experiência como desenvolvedor Java para ambientes corporativos.
- Desde 2016 atua como professor de cursos com soluções open source e Big Data.
- Head de Engenharia de Dados pela Funcional HealthTech
- Founder da Sudoers (<http://blog.sudoers.com.br>)

# Aula 03 - Entendendo as engrenagens da estrutura de dados

# Os números

- **Povo Fenício**
  - **Comércio**
- **Algarismos arábicos**
  - **base 10**
- **base 12**
- **Povo Assírio**
  - **base 60**
- **Computação**
  - **base 2 ou sistema numérico binário**

# Bits e Bytes

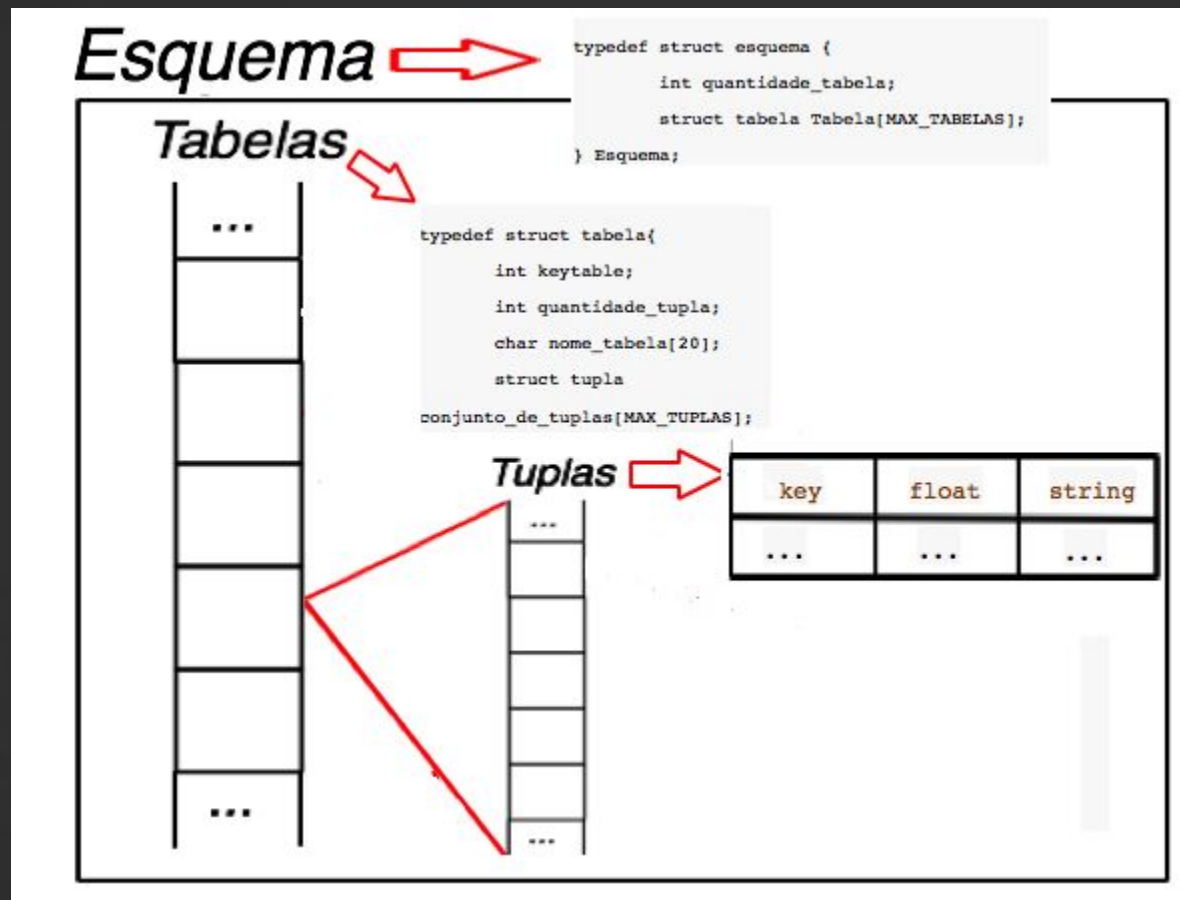
- Como o sistema binário funciona?
  - Tipificação
- Tabela ASCII
  - <http://manpages.ubuntu.com/manic/pt/man7/ascii.7.html>
- Hexadecimal

0	0	6	6	C	12
1	1	7	7	D	13
2	2	8	8	E	14
3	3	9	9	F	15
4	4	A	10		
5	5	B	11		

wikiHow

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2 <sup>63</sup>	0	2 <sup>63</sup> -1	2 <sup>64</sup> -1

- Tuplas





# Tamanho do bloco

- **Formato de Linha**
- **Padrão 8kb**
  - **Transacional**
- **16kb ou 32kb**
  - **DataWarehouses**



# Relações custosas

ID	Nome	Brinquedo	Brinquedo = ID	ID	Descrição	Cor
1	Cliente 1	B1		B1	Carrinho de plástico	Preto
2	Cliente 2	B2		B2	Boneca	NA
3	Cliente 3	B3		B3	Bola	Branca
4	Cliente 4	B4		B4	Corda	NA



# Banco de Dados Documentos

Os bancos de dados orientados a documentos são uma das principais categorias de bancos de dados NoSQL. Os bancos de dados de gráficos são semelhantes, mas adicionam outra camada, o relacionamento, o que lhes permite vincular documentos para uma passagem rápida.

Os bancos de dados orientados a documentos são inerentes aos banco de dados chave-valor. A diferença reside na forma como os dados são processados; em uma banco de dados chave-valor, os dados são considerados inerentemente para o banco de dados, enquanto um sistema orientado a documentos depende da estrutura interna no documento para extrair metadados que o mecanismo de banco de dados usa para otimização adicional.

# Banco de Dados Documentos

Os bancos de dados de documentos armazenam todas as informações para um determinado objeto em uma única instância. Isso faz com que o mapeamento de objetos no banco de dados seja uma tarefa simples, eliminando o mapeamento objeto-relacional. Isso torna-os atraentes para aplicativos web, que estão sujeitas a mudanças contínuas e onde a velocidade é uma questão importante.

```
JSON Object → {  
    "company": "mycompany",  
    "companycontacts": {  
        "phone": "123-123-1234",  
        "email": "myemail@domain.com"  
    },  
    "employees": [  
        {  
            "id": 101,  
            "name": "John",  
            "contacts": [  
                "email1@employee1.com",  
                "email2@employee1.com"  
            ]  
        },  
        {  
            "id": 102,  
            "name": "William",  
            "contacts": null  
        }  
    ]  
}
```

String Value

Object Inside Object

JSON Array

Array Inside Array

Number Value

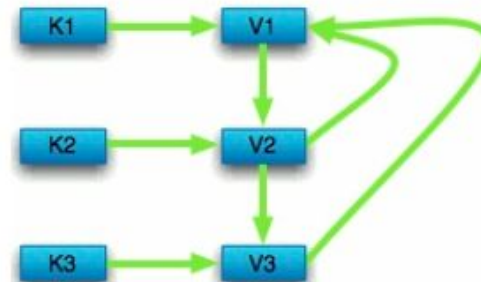
Null Value

# Chave Valor

Optimized for simple look-ups



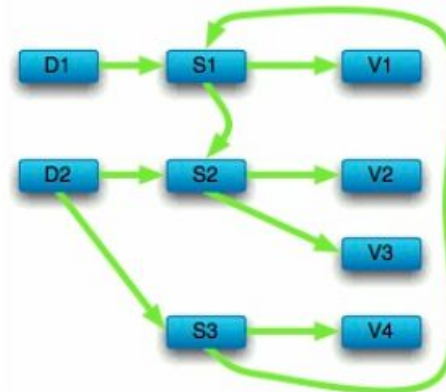
Optimized for traversing connected data



Optimized for “trees” of data



Optimized for seeing the forest and the trees, and the branches, and the trunks



## Tipos de BDs NoSQL



### • Documentos

#### Documento A

```
{
  "titulo": "Este é um post no blog",
  "autor": "Leonardo Eloy",
  "data": "21/04/2009 20:38:00",
  "corpo": "Hoje eu realizei um monte de coisas...",
  "comentarios": [
    {
      "nome": "Josias Cabelo",
      "texto": "Gostei muito do seu post!",
      "data": "28/04/2009 12:31:09"
    },
    {
      "nome": "Fulano de Tal",
      "texto": "Muito interessante, continue",
      "data": "29/04/2009 13:17:31"
    }
  ]
}
```

Fonte: FOWLER e SADALAGE, 2013.

Oracle	MongoDB(documentos)
Instância de Banco de Dados	Instância MongoDB
Esquema	Banco de Dados
Tabela	Coleção
Linha	Documento
RowID	_id
Junção	DBRef

- Acesso fácil aos atributos internos do documento
- Uso de visões materializadas para agregar informações ou estabelecer consultas específicas nos agregados
- Possível fazer consulta dos dados dentro do documento no nível de atributo
- **LIMITAÇÃO: Documentos armazenados devem ser de uma mesma coleção.**





# Banco de Dados Documentos

Amazon SimpleDB



CouchDB



**amazon**  
DynamoDB

- Usando sua máquina virtual, siga o passo a passo
  - <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

## 1 Start MongoDB.

You can start the `mongod` process by issuing the following command:

```
sudo systemctl start mongod
```

copy

If you receive an error similar to the following when starting `mongod`:

```
Failed to start mongod.service: Unit mongod.service not found.
```

Run the following command first:

```
sudo systemctl daemon-reload
```

copy

Then run the start command above again.

## 2 Verify that MongoDB has started successfully.

```
sudo systemctl status mongod
```

copy

You can optionally ensure that MongoDB will start following a system reboot by issuing the following command:

```
sudo systemctl enable mongod
```

copy

# MongoDB - Modelagem

- Os documentos em uma única coleção não precisam ter o mesmo conjunto de campos e o tipo de dados de um campo pode ser diferente nos documentos de uma coleção.
- Para alterar a estrutura dos documentos em uma coleção, como adicionar novos campos, remover campos existentes ou alterar os valores dos campos para um novo tipo, atualize os documentos para a nova estrutura.
- Essa flexibilidade facilita o mapeamento de documentos para uma entidade ou um objeto. Cada documento pode corresponder aos campos de dados da entidade representada, mesmo que o documento tenha variações substanciais de outros documentos da coleção.
- Na prática, os documentos em uma coleção compartilham uma estrutura semelhante e você pode impor regras de validação de documento para uma coleção durante as operações de atualização e inserção.

# Estrutura do Documento

- A decisão principal ao projetar modelos de dados para aplicativos MongoDB gira em torno da estrutura dos documentos e de como o aplicativo representa os relacionamentos entre os dados. O MongoDB permite que dados relacionados sejam incorporados em um único documento.
- **Dados Embutidos**
  - Documentos incorporados capturam relacionamentos entre dados, armazenando dados relacionados em uma única estrutura de documento. Os documentos do MongoDB possibilitam embutir estruturas de documentos em um campo ou array dentro de um documento. Esses modelos de dados desnormalizados permitem que os aplicativos recuperem e manipulem dados relacionados em uma única operação de banco de dados.

# Estrutura do Documento

```
{  
  _id: <ObjectId>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```

Embedded sub-  
document

Embedded sub-  
document

# Referências

user document

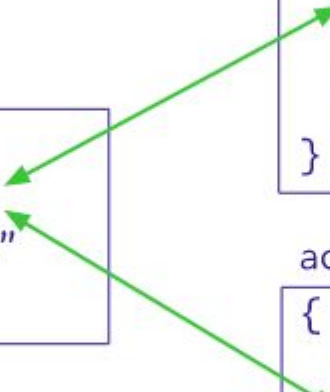
```
{  
  _id: <ObjectId1>,  
  username: "123xyz"  
}
```

contact document

```
{  
  _id: <ObjectId2>,  
  user_id: <ObjectId1>,  
  phone: "123-456-7890",  
  email: "xyz@example.com"  
}
```

access document

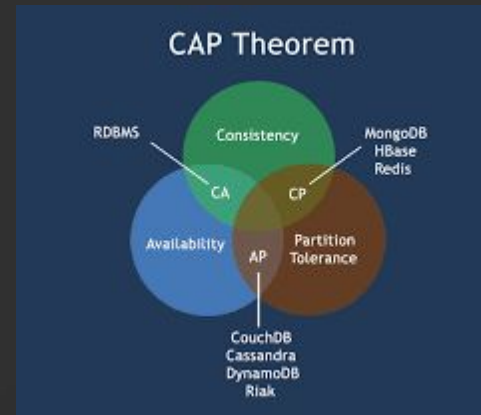
```
{  
  _id: <ObjectId3>,  
  user_id: <ObjectId1>,  
  level: 5,  
  group: "dev"  
}
```



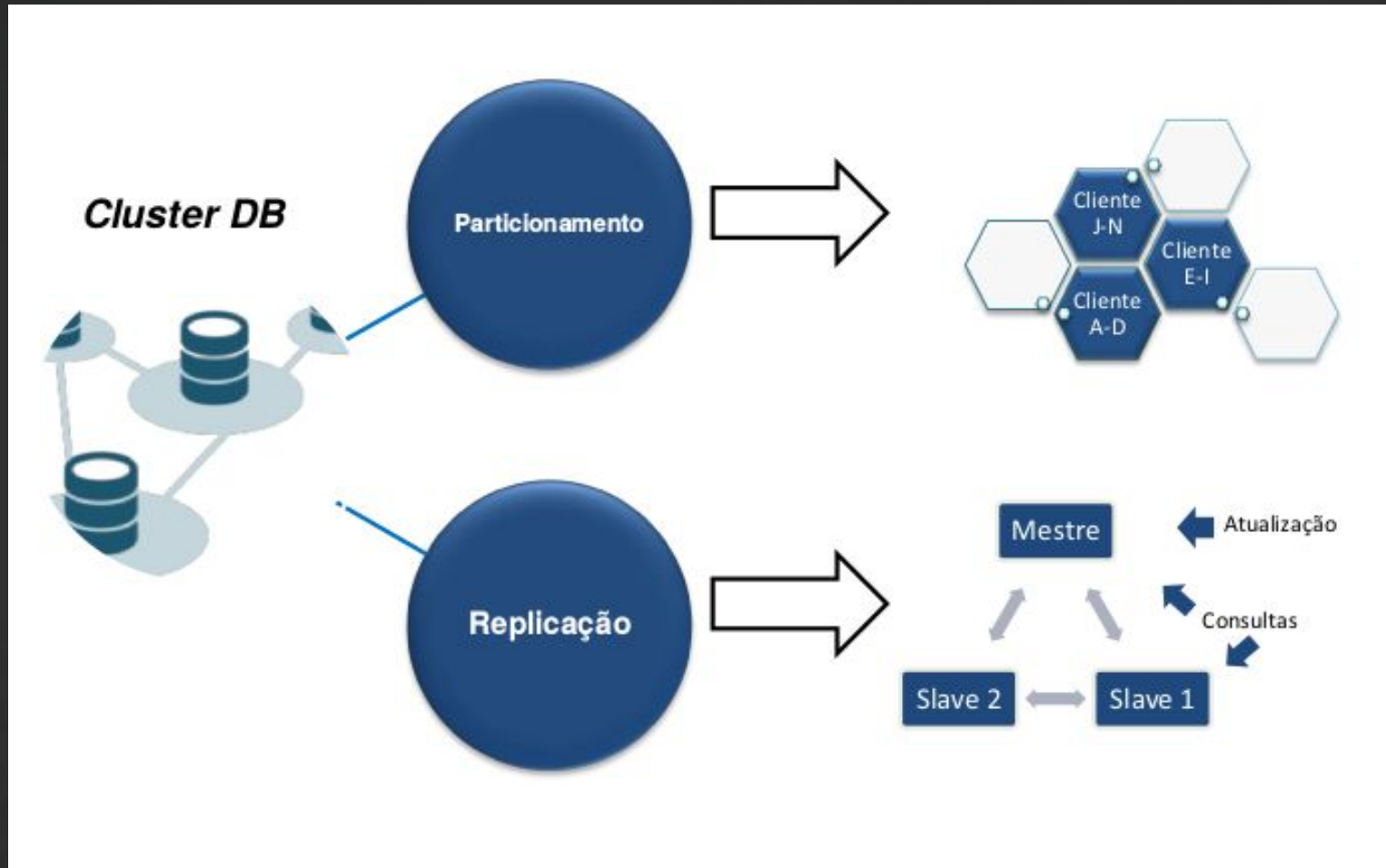


# Lembra do ACID?

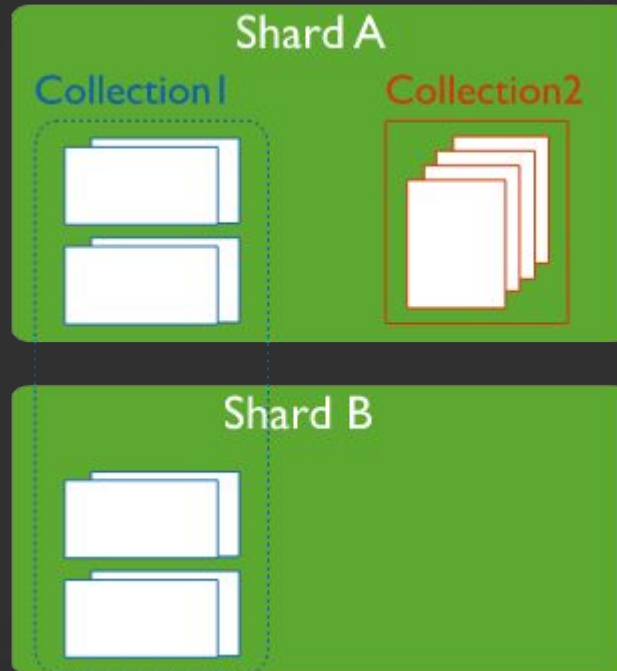
- **Atomicidade**
  - Garante em um simples documento
  - Para múltiplos documentos
    - A partir da versão 4.0
- **Consistência**
  - Vamos olhar no teorema CAP
- **Isolamento**
  - Sendo otimista
- **Durabilidade**
  - Se persiste, com certeza tem.



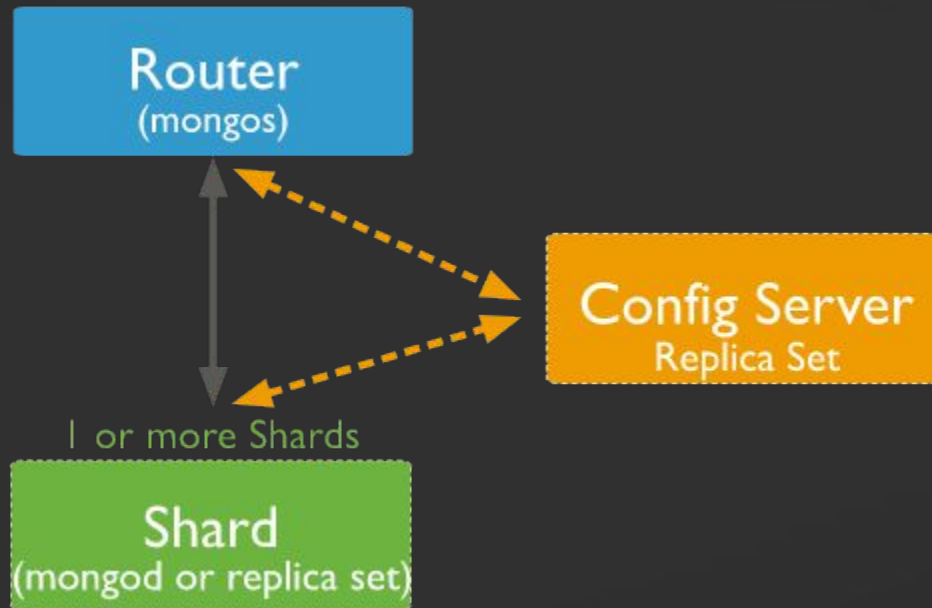
# Lembra do Relacional?



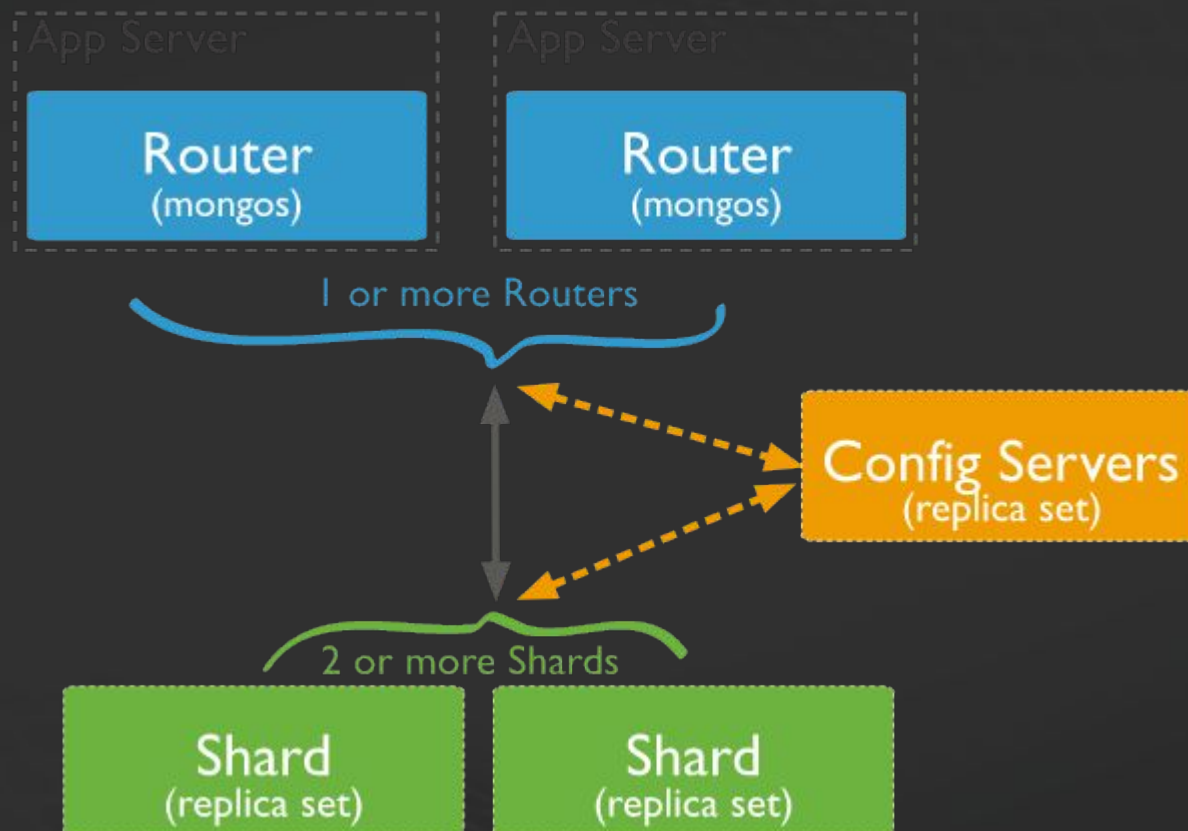
# Shard



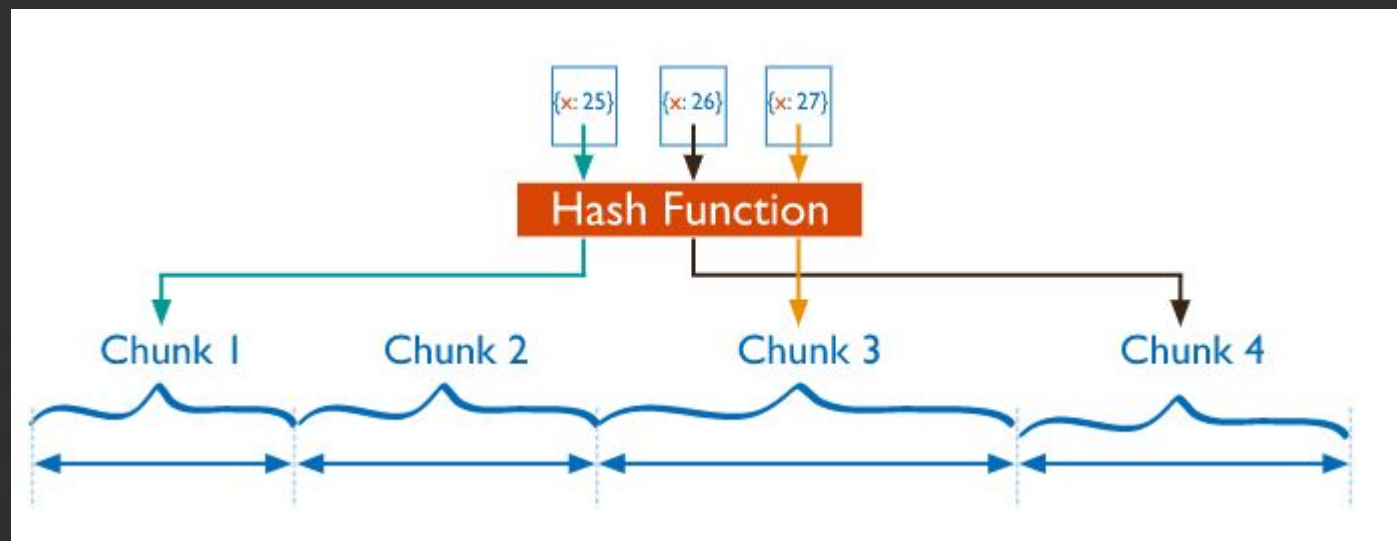
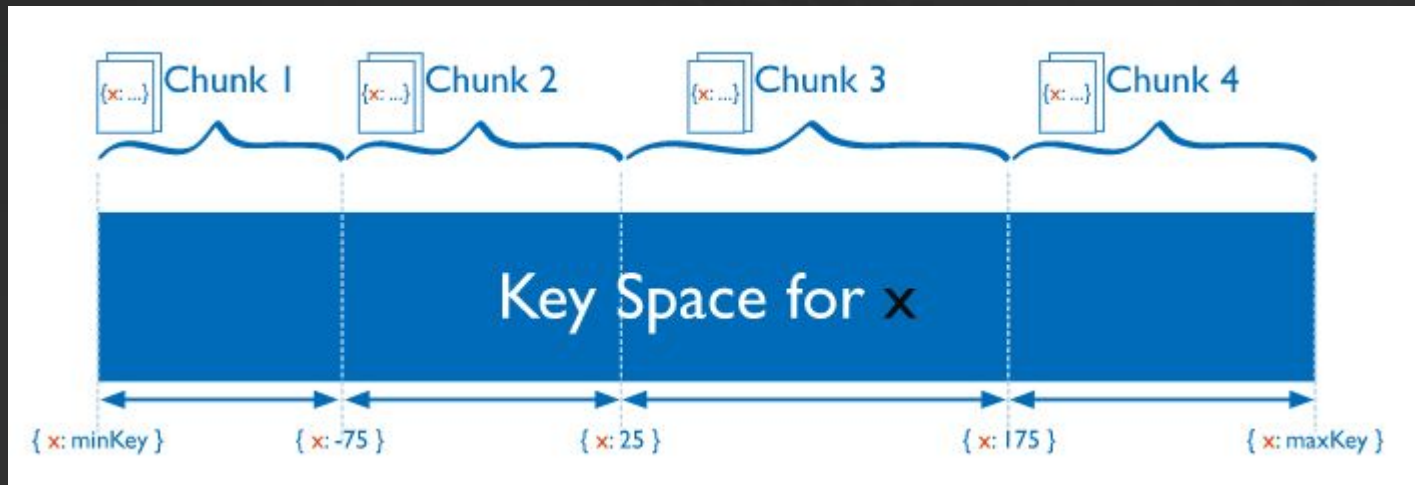
# Arquitetura MongoDB



# Arquitetura MongoDB

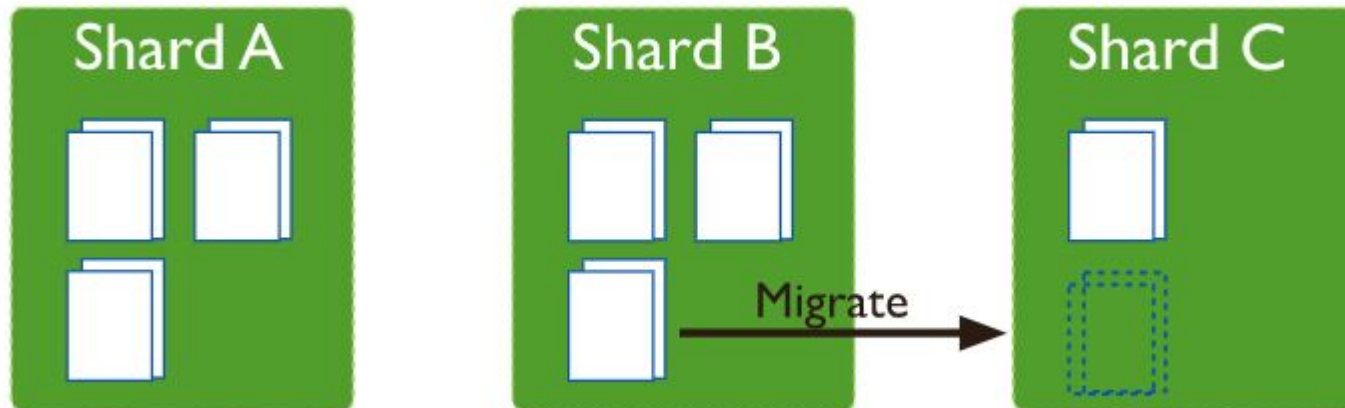


# Chunks





# Balanceamento



# Mais detalhes

1. BSON Documents
  - a. GridFS (<https://docs.mongodb.com/manual/core/gridfs/> )
2. Restrições de Nomes
3. Namespaces
4. Índices
5. Data
6. RéplicaSet/Shards  
(<https://docs.mongodb.com/manual/core/sharding-data-partitioning/>)  
<https://docs.mongodb.com/manual/core/sharded-cluster-shards/>  
<https://docs.mongodb.com/manual/core/sharding-shard-key/>
7. Agregações

Glossário - <https://docs.mongodb.com/manual/reference/glossary/>

- Faça os exercícios do arquivos `mongodb_exercicios.zip`

## BOA NOITE!

Referência no ensino de Tecnologia em **Graduação, Pós, EAD, MBA, cursos livres e treinamentos empresariais**. Lidera iniciativas em Inteligência Artificial, Ciência de Dados, Robótica, Engenharia da Computação, Big Data, UX e Transformação Digital. Criou a incubadora Impacta Open Startup, exclusiva para seus estudantes. É campeã na formação de times para Hackathons, eventos de inovação patrocinados por gigantes: NASA, IBM, Deloitte, Shell, Santander, Itaú, Globo e Fiesp. É uma das marcas mais admiradas pela Comunidade Tech da América Latina. Atua no mercado desde 1988 e formou mais de 1 milhão de pessoas e certificou mais de 25 mil empresas de diversas áreas da Economia. Ver mais em: [impacta.edu.br](http://impacta.edu.br) e [impacta.com.br](http://impacta.com.br)

