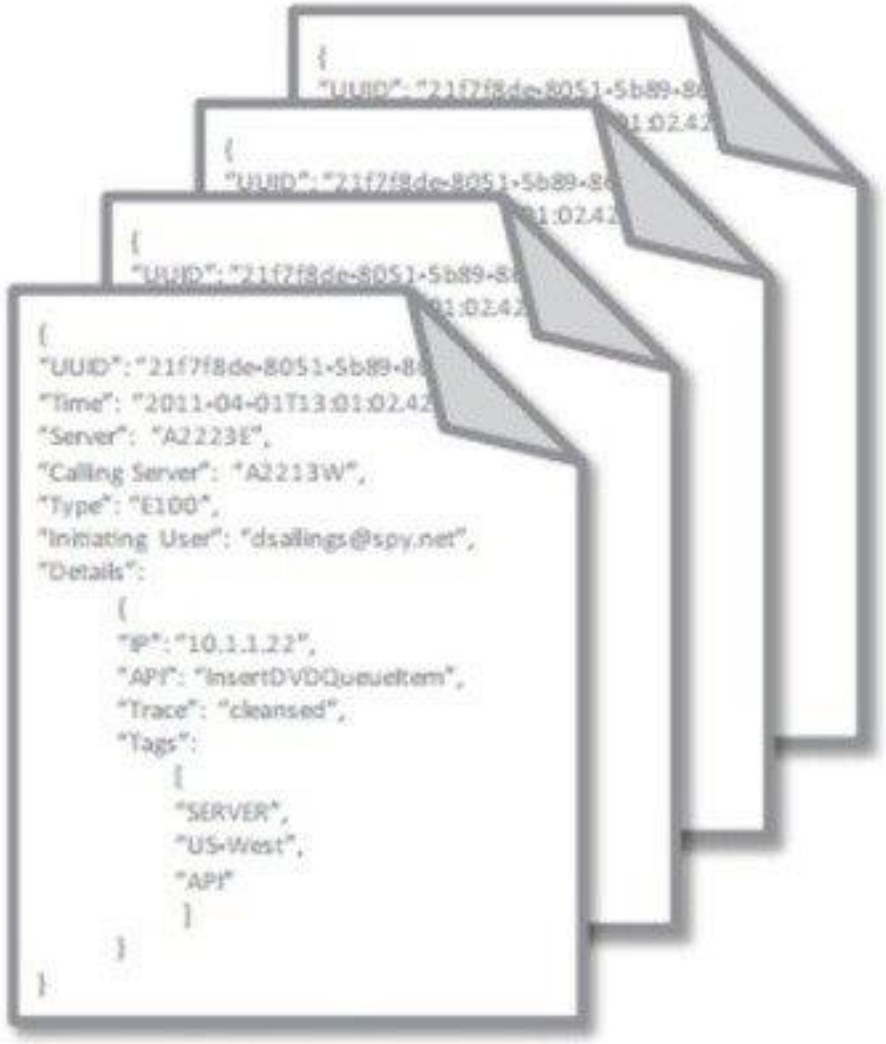
NoSQL
Orientado a documentos

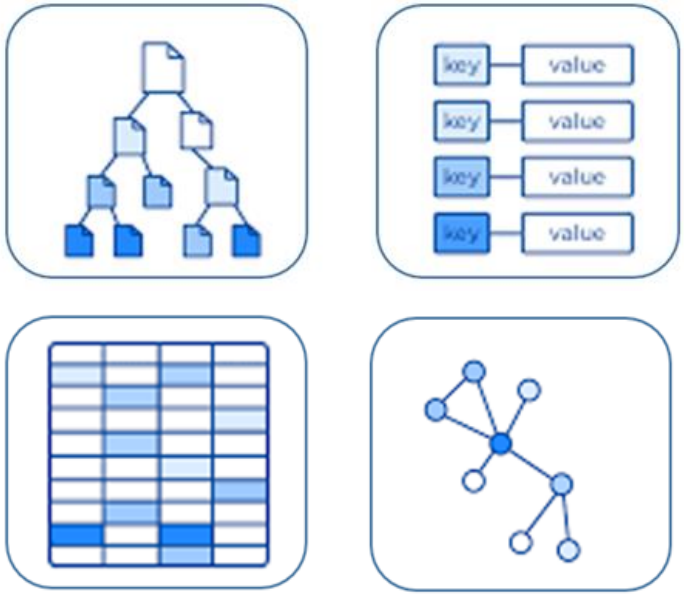


Profa. Carolina Torres

INSTITUTO FEDERAL
Pernambuco

Modelos de Dados

- Modelos de Dados mais comuns:
  - Chave-valor
  - **Orientado a Documentos**
  - Orientado a Colunas
  - de Grafo

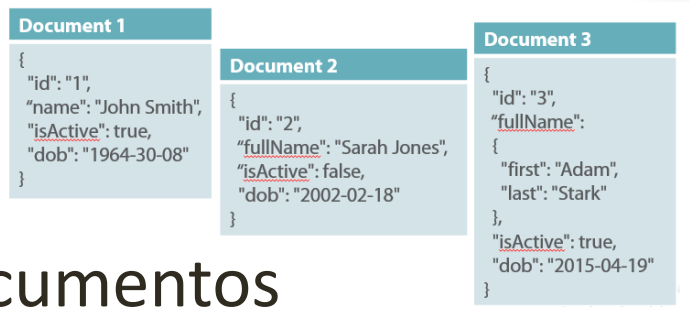


INSTITUTO FEDERAL
Pernambuco

Documentos

- Cada documento deveria ser autossuficiente
  - Deve apresentar todas as ‘entidades’ e ‘relacionamentos’ condensados
  - Indicado que contenha toda a informação necessária para não serem necessários JOINs

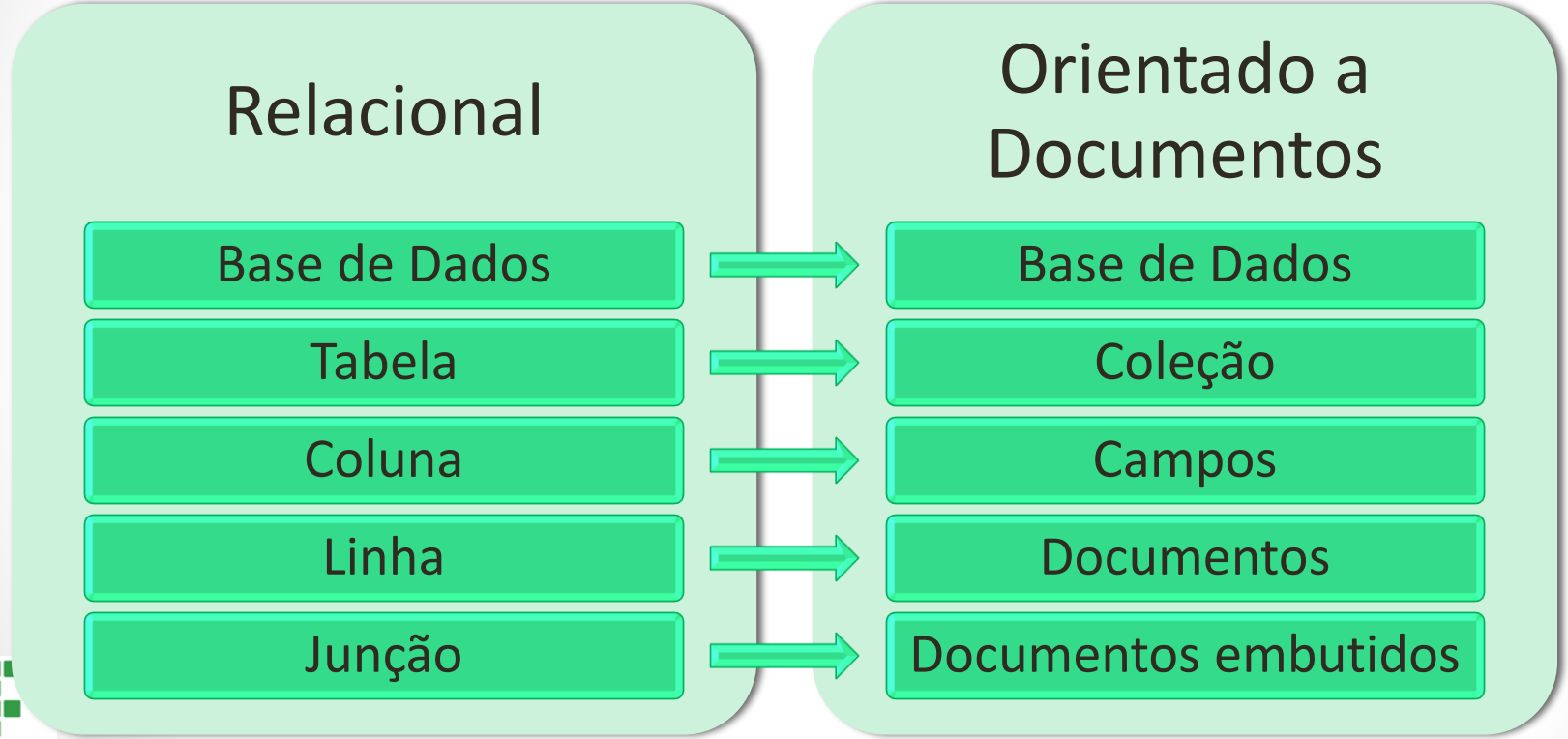
Orientado a Documentos



- Armazena os dados em documentos semi-estruturados
  - Exemplo: JSON, XML, ...
- Cada documento pode ser uma unidade completa da informação, permitindo distribuição em múltiplos servidores
  - Mas permitindo referências entre documentos também

INSTITUTO FEDERAL
Pernambuco

Terminologia



Orientação a Documentos

Prós

- Sem esquema fixo
  - Esquema é definido em execução
  - Indica-se o mínimo de planejamento para otimização
- Facilidade em aprender/utilizar
  - Por utilizar tecnologia JSON
- *MapReduce*
  - Trabalhar com *Big Data*

Contras

- Redundância
  - Mesmo dado pode estar em dois ou mais documentos
- Sem JOINs
  - Performance cai se precisar *linkar* documentos
- Ausência de transações
  - Não garante ACID (atomicidade, consistência, isolamento e durabilidade)

INSTITUTO FEDERAL
Pernambuco

Prática

NoSQL de Documentos

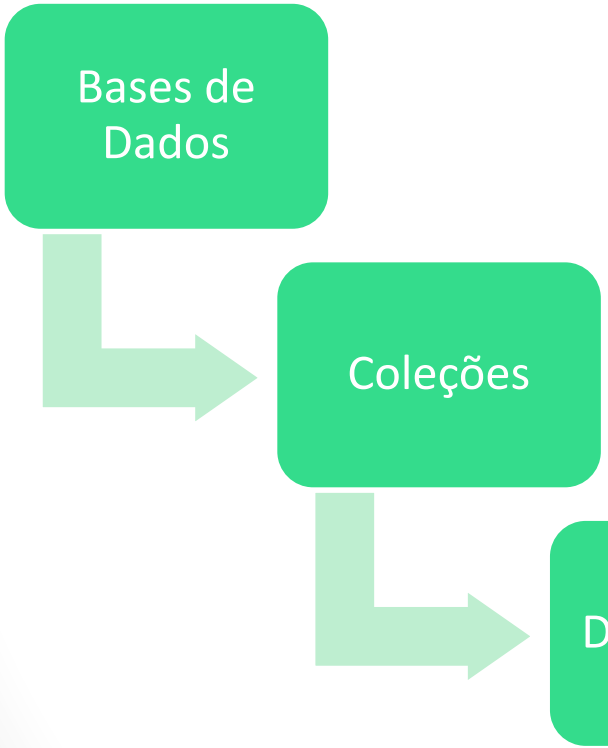
Utilizando o MongoDB

MongoDB

- C:\Arquivos de Programas\MongoDB\server\5.0\bin\mongo.exe
- Escrito em C++
- Multiplataforma: Windows, Linux e Mac...
- Download gratuito em https://www.mongodb.com/try/download/community
- Ferramenta CASE: https://robomongo.org/

INSTITUTO FEDERAL
Pernambuco

MongoDB



Formato BSON, semelhante ao JSON

INSTITUTO FEDERAL
Pernambuco

MongoDB

Quem usa?

- Google
- Ebay
- Facebook
- CISCO
- SAP
- Telefonica
- ...

Quando usar?

- Big Data
- Alta escalabilidade
- Esquema instável
- Busca simples
- Dados descritivos

Concorrentes

- CouchDB
- Couchbase
- RavenDB
- ...

Operações Básicas

Database e Collections, CRUD

Operações Básicas
Databases

use <*database name*>	• Trocar para a base de dados fornecida ao commando, se ela não existir será criada.
db	• Checar qual a base de dados atualmente selecionada
show dbs	• Exibir a lista de bases de dados
db.dropDatabase()	• Apagar a base de dados em uso

INSTITUTO FEDERAL
Pernambuco

Operações Básicas
Collections

db.createCollection (name)	• Criar uma coleção
db.*collectionname*.insert (*document*)	• Não é necessário criar uma coleção, basta inserir nela que o MongoDB cria automaticamente a coleção
show collections	• Listar os nomes de todas as coleções da base de dados selecionada
db.*collectionname*.drop ()	• Apaga uma coleção da base de dados

Antes de falar do CRUD...

- Todo documento tem um identificador único
  - **Nome:** _id

Pode ser definido explicitamente na hora de inserir um documento

```
{
    _id: 1,
    nome: 'Carolina'
    idade: 33
}
```

Ou será definido automaticamente pelo sistema (ObjectId)

```
{
    nome: 'Josino',
    idade: 34
}
```

INSTITUTO FEDERAL
Pernambuco

Documentos

- MongoDB aceita vários tipos de dados aceitos:
  - Strings, números (inteiros ou decimais), booleanos, arrays, listas, datas, Null, subobjetos...

Customer Document

```
"customer" =
{
   "id": "Customer:1",
   "firstName": "John",
   "lastName": "Wick",
   "age: 25,
   "address": {
        "country": "US",
        "city": "New York",
        "state": "NY"
        "street": "21 2nd Street",
      },
   "hobbies": [ Football, Hiking ],
   "phoneNumbers": [
      {
        "type": "Home",
        "number": "212 555-1234"
      },
      {
        "type": "Office",
        "number": "616 565-6789"
      }
    ]
}
```

INSTITUTO FEDERAL
Pernambuco

Operações CRUD

- CREATE (Insert)
- RETREAVE (Find)
- Update
- DELETE

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Insert

- Para inserir, há dois métodos:
  - insertOne()
  - insertMany()

```
> db.collectionname.insertOne(documento)
```

```
> db.collectionname.insertMany([documento1, documento2])
```

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Insert

- **Exemplos:**
  - Inserir documento sobre a professora Carol (sem *_id*)

    ```
    > db.profs.insertOne( { nome: "Carol", idade: 33 } )
    ```

  - Inserir documento sobre a professora Maria (definindo o *_id* e subobjeto)

    ```
    > db.profs.insertOne( { _id: 23, nome: "Maria", idade: 40, Endereco:
    {rua:"Rua teste", numero:456} } )
    ```

  - Inserir dois documentos (um sem e outro com _id definido)

    ```
    > db.profs.insertMany(
    [{ nome: "Torres", idade: 30 }, { _id: 716, nome: "Silva", idade: 45 } ])
    ```

INSTITUTO FEDERAL
Pernambuco

Operações CRUD

- CREATE (Insert)
- RETREAVE (Find)
- Update
- DELETE

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Find

- Para selecionar, há apenas o método find()

```
> db.collectionname.find()
```

- Para realizar filtros:

```
> db.collectionname.find
({ campo : valor })
```

- Para realizar projeção:

```
> db.collectionname.find
({ }, { campoNÃO:0,campoSIM:1 })
```

Relacional	MongoDB
=	$eq
>	$gt
>=	$gte
<	$lt
<=	$lte
!=	$ne
IN (utilizado em arrays)	$in
NOT IN (utilizado em arrays)	$nin

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Find

- **Exemplos (filtros):**
  - Buscar os dados de todos os professores

    > db.profs.find ()
  - Buscar os professores com idade igual a 40

    > db.profs.find({Idade : 40})
  - Buscar os professores com idade maior ou igual a 30 e menor ou igual a 45

    > db.profs.find({idade : { $gte:30, $lte:45} })
  - Buscar os professores com 30 anos e que morem no número 456.

    db.profs.find({"Endereco.numero":456, Idade:30})

    Obs: acesso a subobjetos é feito utilizando o ponto e o AND é realizado utilizando vírgula

Operações CRUD - Find

- **Exemplos (projeção):**
  - Buscar apenas nomes e _id dos professores

    > db.profs.find ({ } , {nome:1})

    Obs: especificar 1 para as colunas que devem ser retornadas, e 0 para as que não devem. _id é retornado por padrão.
  - Buscar nome do professor cujo _id seja 23

    > db.profs.find ({_id:23},{nome:1,_id:0})

Operações CRUD - Find

- **Exemplos (ordenação, contagem, limitação):**
  - Trazer os nomes dos professores ordenados pelo _id

    > db.profs.find ({},{nome:1,_id:0}).sort({_id: 1})

    Obs: especificar 1 para ordem crescente e -1 pra ordem decrescente.
  - Quantos professores estão cadastrados?

    > db.profs.count()
  - Qual o primeiro professor cadastrado que possui menos de 35 anos?

    > db.profs.findOne({Idade:{$lte:35}},{_id:0})
  - Retornar apenas os 3 primeiros professores

    > db.profs.find().limit(3)
  - Retornar todos os professores exceto os dois primeiros

    > db.profs.find().skip(2)

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Find

- **Exemplos (LIKE e DISTINCT):**
  - Quais os professores cujo nome comece com M
    > db.profs.find({nome:/^M/})
  - Quantos professores tem a letra a no seu nome?
    > db.profs.find({nome:/a/}).count()
  - Quais os professors que possuem seu nome terminando com s?
    - db.profs.find({nome:/s$/})
  - Quais as idades dos professores?
    - db.profs.distinct("Idade")

INSTITUTO FEDERAL
Pernambuco

Operações CRUD

- CREATE (Insert)
- RETREAVE (Find)
- Update
- DELETE

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Update

- Para atualizar existem 3 métodos:
  - updateOne( )
  - updateMany( )
  - replaceOne( )
- Podem receber 3 parâmetros:
  1. Filtros
  2. Atualização
  3. Opções (verificar aqui)

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Update

- **Exemplos:**
  - Atualizar a idade para 50 do professor com _id 23

    > db.profs.updateOne({ _id: 23 }, { $set: {Idade: 50}})
  - Criar a informação das disciplinas ministradas pelo professor de _id 23

    > db.profs.updateOne({ _id: 23 }, { $set:{disciplinas: ['BD','BD1','BD2']}})
  - Apagar a informação de idade dos professores

    > db.profs.updateMany({},{ $unset:{idade: ""}})
  - Incluir o status ativo em todos os professores?

    > db.profs.updateMany({ }, { $set:{status: "ativo"}})

INSTITUTO FEDERAL
Pernambuco

Operações CRUD

- CREATE (insert)
- RETREAVE (find)
- Update
- DELETE

Operações CRUD - Delete

- Para deleção existem 2 métodos:
  - deleteOne( )
  - deleteMany( )

```
> db.collectionname.deleteOne ( {filtros} )
```

```
> db.collectionname.deleteMany ( {filtros} )
```

INSTITUTO FEDERAL
Pernambuco

Operações CRUD - Delete

- **Exemplos:**
  - Deletar todos os professores

    > db.profs.deleteMany ({})
  - Deletar todos os professores com 40 anos

    > db.profs.deleteMany ({idade:40})
  - Deletar o primeiro professor com mais de 25 anos?

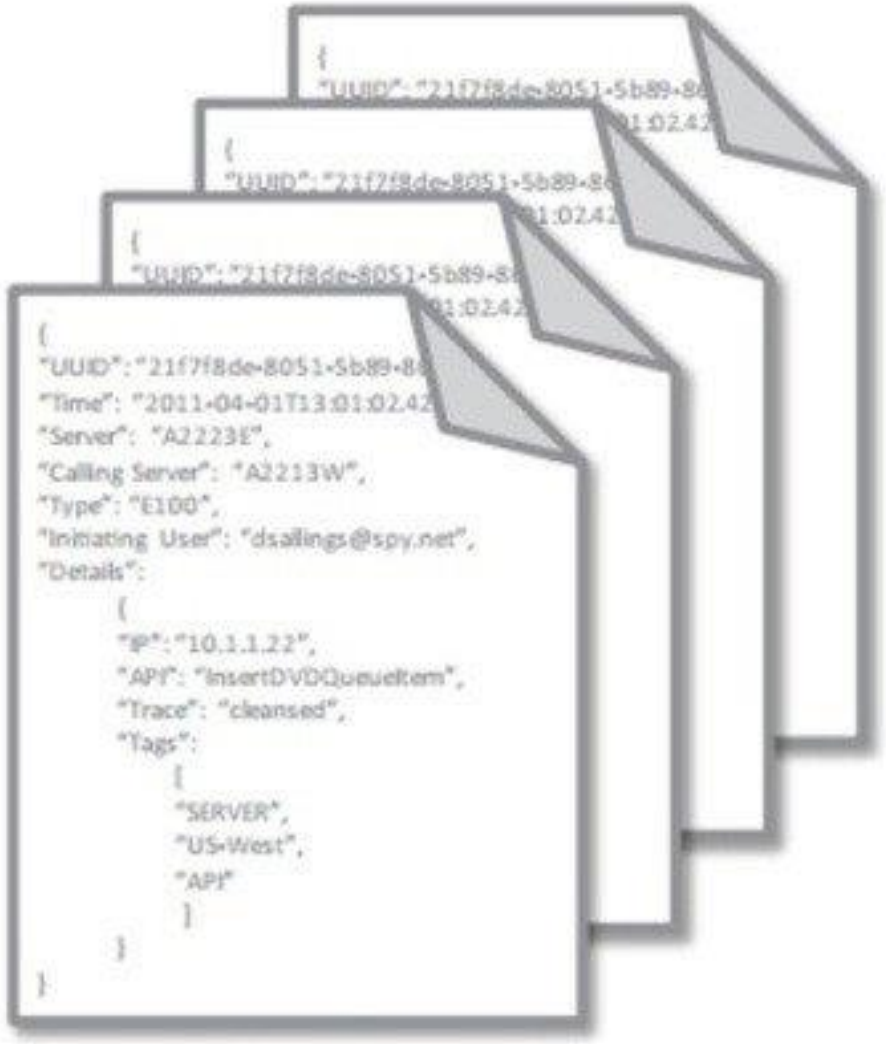
    > db.profs.deleteOne ({idade:{$gte:25}})

Links interessantes

- Manual oficial:
  - Completo: https://docs.mongodb.com/manual
  - Overview: https://docs.mongodb.com/manual/core/databases-and-collections/
  - CRUD: https://docs.mongodb.com/manual/crud/
- Exemplo prático:
  - https://www.youtube.com/watch?v=j-cjF1GkEMQ&t=2s

INSTITUTO FEDERAL
Pernambuco

NoSQL
Orientado a documentos



Profa. Carolina Torres

INSTITUTO FEDERAL
Pernambuco