

TECNOLOGIAS DE PROGRAMAÇÃO DE SISTEMAS DE
INFORMAÇÃO

MongoDB

SISTEMAS GESTORES DE BASES DE DADOS II | Prof. Magno
Andrade | Prof. Jorge Louro

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Conteúdo da apresentação**
 - Replicação



REPLICAÇÃO

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Introdução**

- Um *replica set* é um grupo de processo ***mongod*** que mantém o mesmo conjunto de dados.
- Estes disponibilizam redundância e alta disponibilidade.
- São utilizados para os desenvolvimentos de produção.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Introdução**

- Com várias cópias de dados em diferentes servidores, a replicação fornece um nível de tolerância a falhas contra a perda de um servidor.
- Em alguns casos, a replicação pode oferecer uma maior capacidade de leitura, pois os clientes podem enviar operações de leitura para diferentes servidores.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

● **Replica Set - Membros**

○ Tem os seguintes membros:

■ Primário.

- É o único que recebe todas as operações de escrita.

■ Secundários.

- Estes replicam as operações do primário para manter um conjunto de dados idêntico.
- Possível definir para um objectivo específico, por exemplo, nunca tornar-se um primário ou iniciar eleições; nunca permitir operações de leitura.

■ Árbitro.

- Não podem tornar-se um primário.
- Não guardam dados, apenas são usados para as eleições do novo primário.
- Tem apenas 1 voto na eleição.
- Nunca criar um árbitro num mesmo sistema, com um primário ou secundário.



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Replicação no *MongoDB***

- Um *replica set* contém vários nós e opcionalmente um nó árbitro.
- Destes nós, apenas um é o nó primário (***primary***), enquanto os outros são nós secundários (***secondary***).
- O nó primário recebe todas as operações de escrita.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

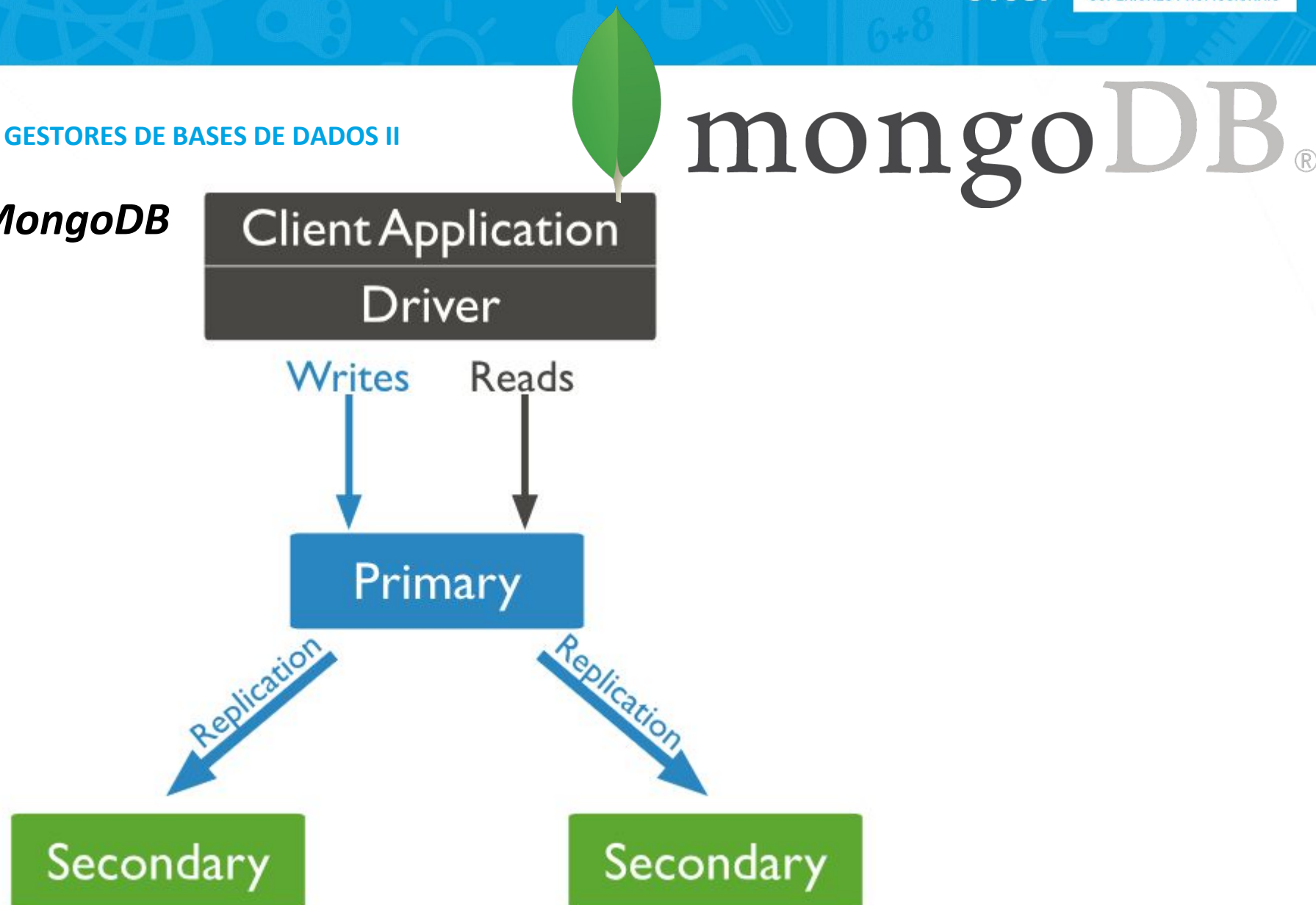
- **Replicação no *MongoDB***



- Todas as operações efectuadas ao conjunto de dados do nó primário são guardadas num *log* de operação (***oplog***).
- O ***oplog*** é uma colecção *capped* especial, que mantém um registo de todas as operações que modificaram os dados guardados na base de dados.
- Todos os membros do *replica set* tem uma cópia do ***oplog***, para manterem os estado actual da base de dados.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Replicação no *MongoDB*



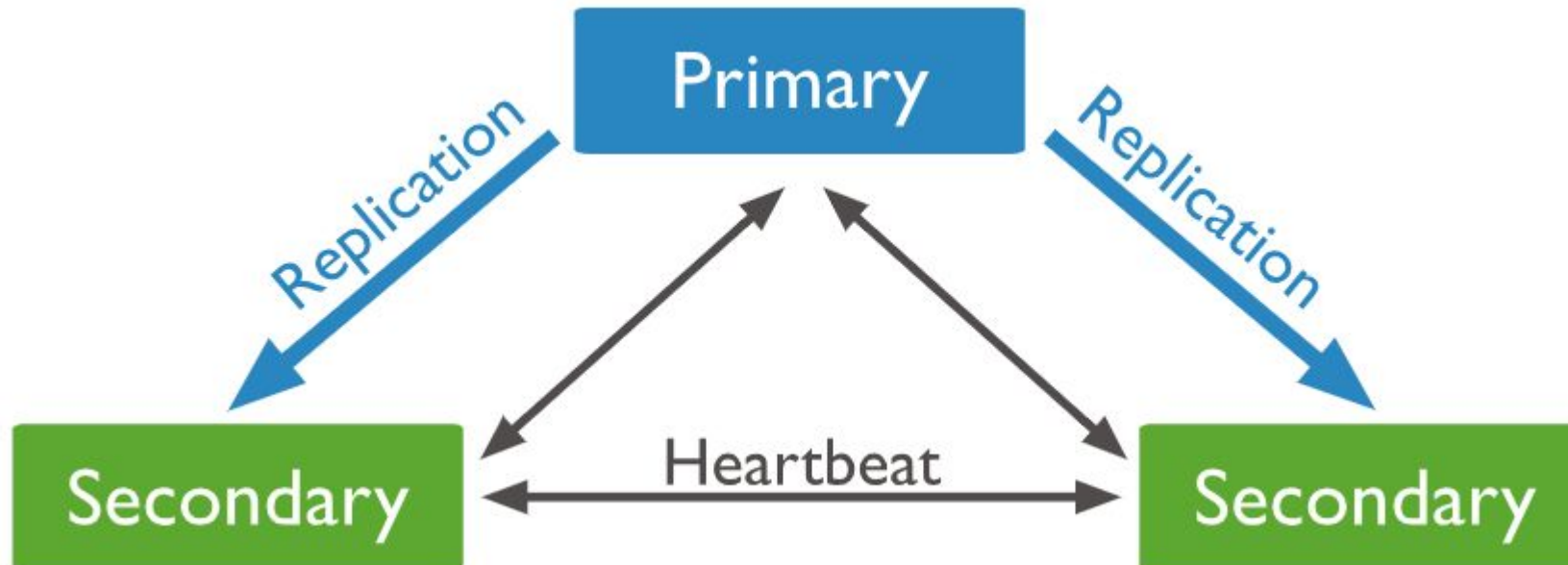
UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Replicação no *MongoDB***

- Os secundários replicam o **oplog** do primário, para reflectir as mudanças nos dados realizadas no primário.
- Se o primário não estiver disponível, um secundário elegível realizará uma eleição para eleger-se o novo primário.

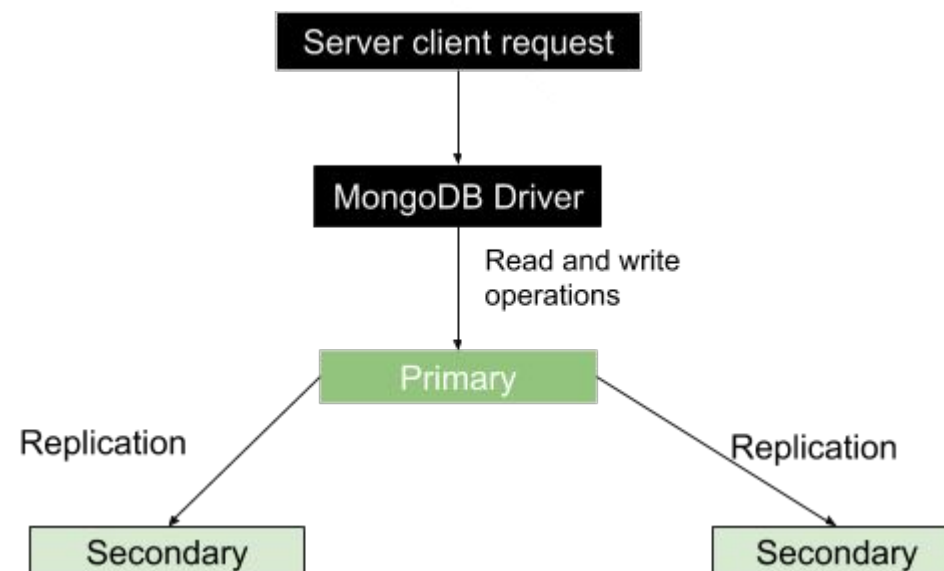


UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Replicação assíncrona**

- Os secundários aplicam às operações do primário de forma assíncrona. Ao aplicar operações depois do primário, os conjuntos podem continuar funcionando apesar da falha de um ou mais membros.



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Falha automática**

- Quando o primário não comunica com outros membros, mais do que o tempo configurado (***electionTimeoutMillis*** - 10 segundos por omissão):
 - Um nó secundário pede uma eleição para ser o novo primário.
 - O cluster tenta completar a eleição do novo primário e voltar as operações normais.
- O tempo para a eleição normalmente não deve exceder os 12 segundos.
- A latência da rede é um factor importante na eleição:
 - Se for grande, a eleição demora mais tempo a completar, isto significa que o sistema fica sem primário durante mais tempo.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Falha automática**

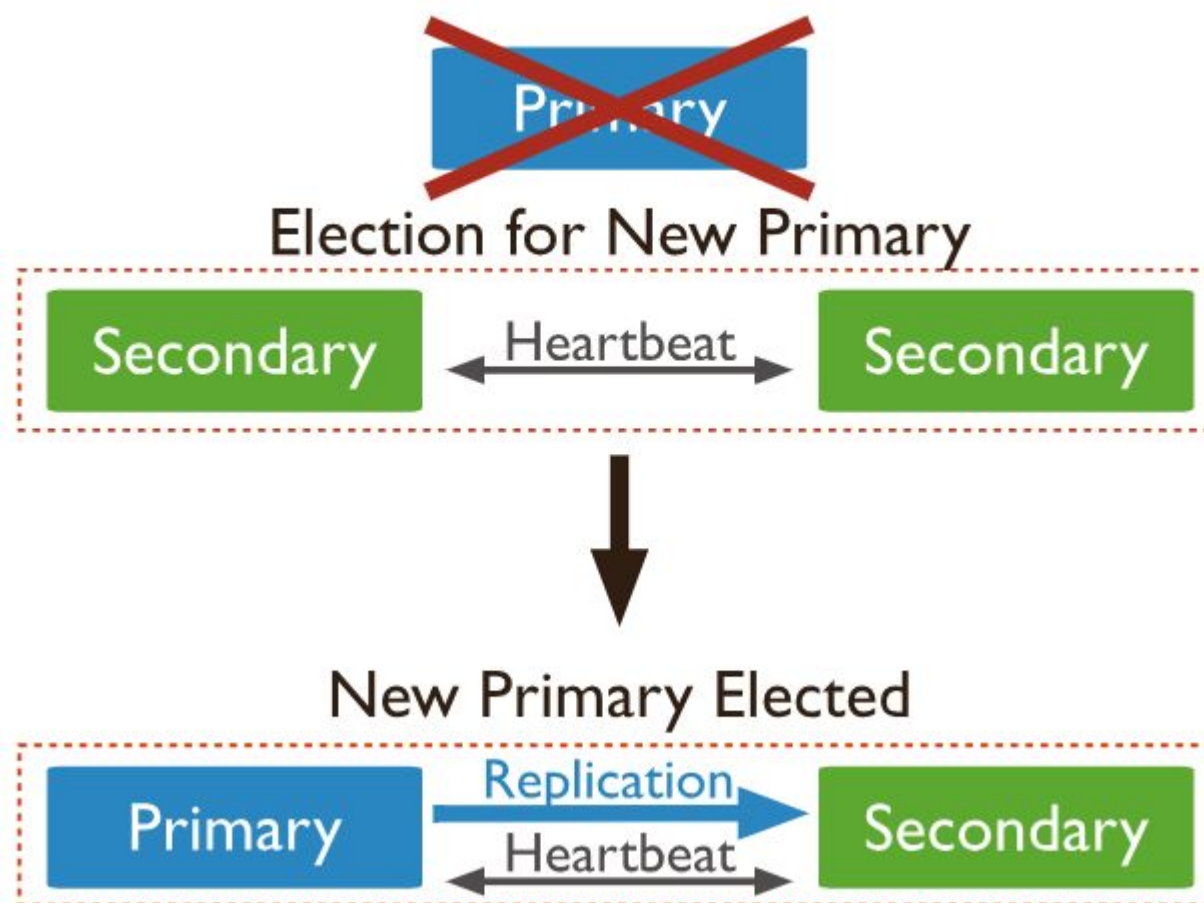
- Enquanto não acabar a eleição:
 - Não pode processar operações de escrita.
 - Pode continuar a servir operações de leitura, se estiver configurado que estas podem ser realizadas nos secundários enquanto o primário está *offline*.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Falha automática



mongoDB®



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Operações de leitura**

- Realizadas no nó primário, por omissão.
- Podem ser configuradas para ser utilizados os nós secundários.
- Se utilizados os nós secundários pode acontecer o seguinte:
 - Dado a replicação dos dados ser assíncrona, os dados retornados podem não ser os mais recentes.



- **Operações de leitura**

- Transacções multi-documento que contenham operações de leitura têm de utilizar o nó primário.
- Todas as operações numa dada transacção devem ser encaminhadas para um mesmo membro.

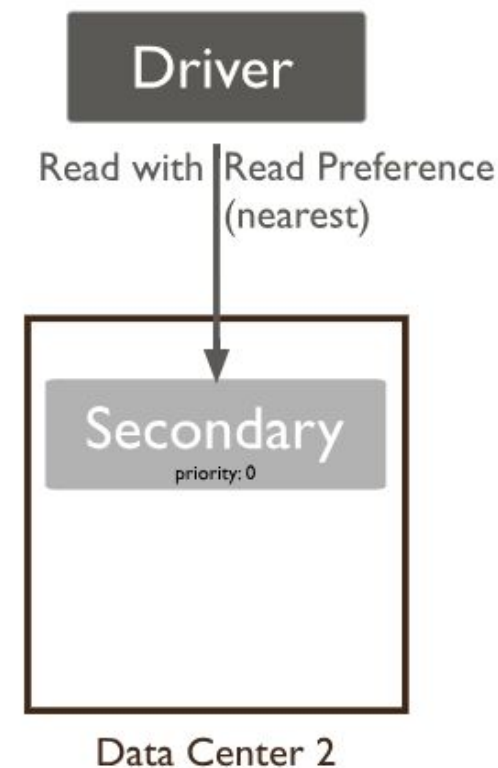
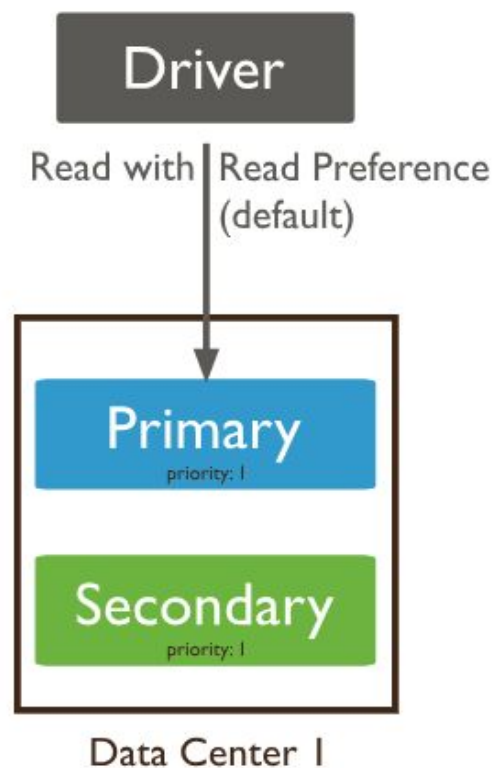
UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Preferência de leituras**

- Descreve como o *MongoDB* encaminha os pedidos de leitura do utilizador para os membros do *replica set*.



mongoDB®



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Preferência de leituras**

- Existem os seguintes modos:

- *primary*

- por omissão, todas as operações de leitura são realizadas no primário.

- *primaryPreferred*

- as operações são realizadas no primário, mas se este encontrar-se indisponível são efectuadas nos secundários.

- *secondary*

- todas as operações são realizadas nos membros secundários.

- *secondaryPreferred*

- as operações são realizadas nos secundários, mas se estes encontrarem-se indisponíveis são efectuadas no primário.

- *nearest*

- as operações são efectuadas no membro com menos latência de rede, independente do tipo (primário ou secundário).

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Preferência de leituras**

- Estes modos são apenas efectuados num objecto de ligação para controlar o fluxo de leituras.
- Seguinte comando:
 - `db.getMongo().setReadPref('primaryPreferred')`
- Outras opções:
 - `db.getMongo().setSlaveOk()` → para a sessão actual permitir operações de leitura no secundário.
 - `db.getMongo().slaveOk()`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Replica set - eleições**

- Os **replica set** usam as eleições para escolher o novo membro que irá ser o primário.
- É despoletado nas seguintes situações:
 - na adição de um novo nó ao *set*.
 - na inicialização do *set*. (ex: comando `rs.initiate()`)
 - na manutenção de membros do *set*. (ex: comandos `rs.stepDown()` ou `rs.reconfig()`)
 - os secundários perdem conexão com o primário.
- Num *replica set*, só podem **existir 7** membros votantes.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- ***Replica set* - eleições**
 - *Heartbeats*
 - Os membros do *replica set* enviam *pings* uns para os outros, a cada dois segundos.
 - Se este *ping* não retornar dentro de 10 segundos, o membro é marcado como membro inacessível.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Replica set - eleições**
 - Prioridade dos membros
 - A prioridade afecta o tempo e o resultado das eleições.
 - Os secundários com maior prioridade, chamam eleições primeiro que os de menor prioridade e tem uma probabilidade maior de ser o primário.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- ***Replica set* - eleições**

- Prioridade dos membros

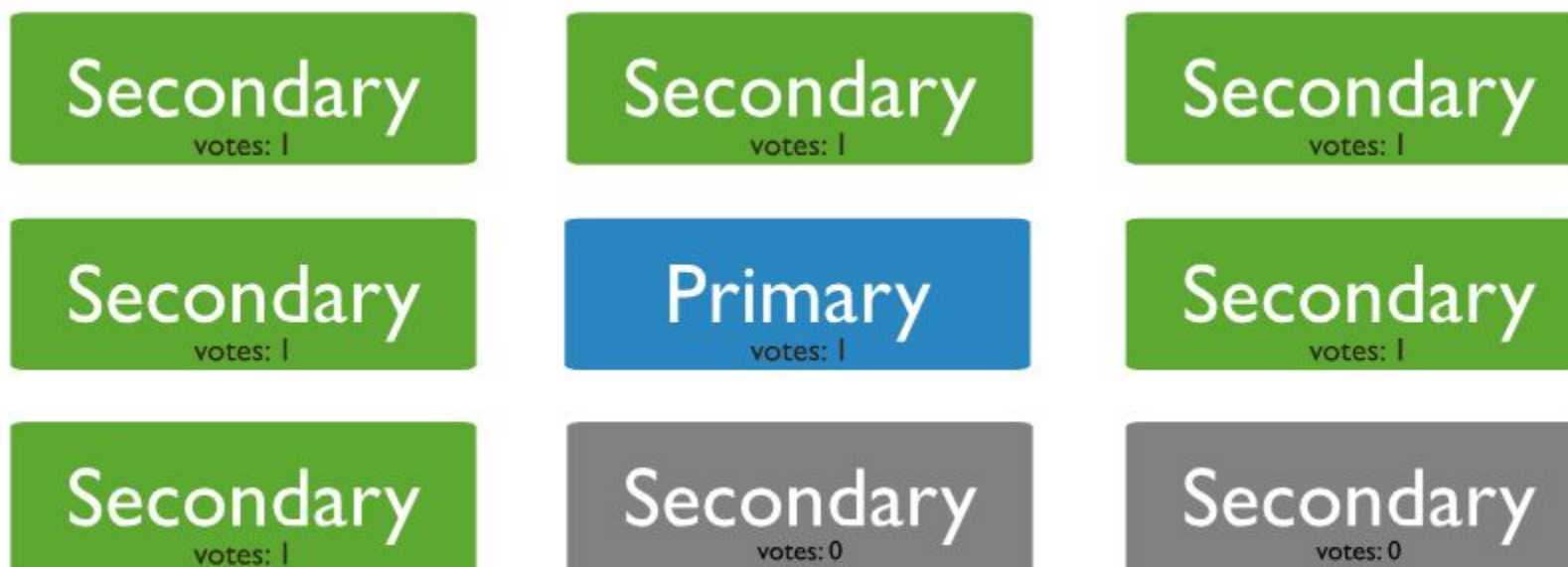
- No entanto, um de menor prioridade pode ser eleito o primário por breves momentos, mesmo existindo um de maior prioridade.
- Os membros do *replica set* continuam a chamar eleições, até que o membro com a maior prioridade seja eleito o primário.
- Membros com a prioridade 0, não podem ser o primário.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Replica set - eleições**
 - Membros não votantes
 - Tem a prioridade 0.
 - Não votam mas guardam os dados.



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- ***Replica set*** - eleições
 - Membros não votantes - Exemplo



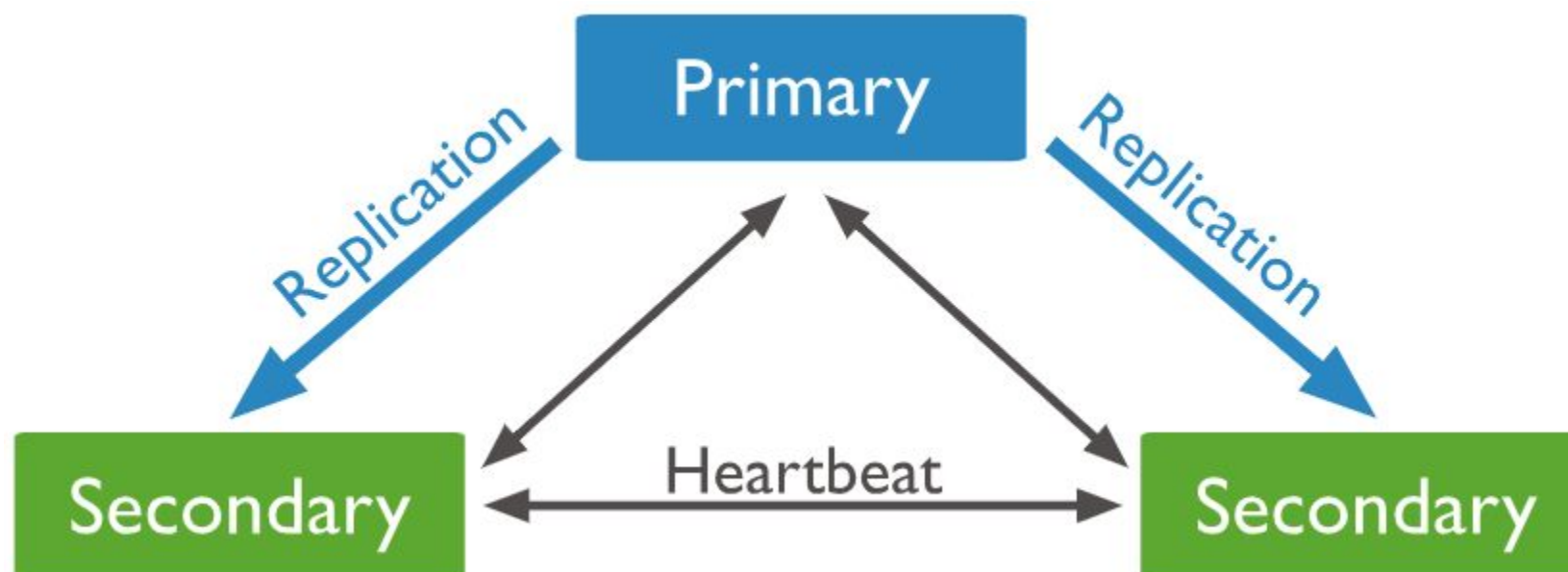
mongoDB®

```
{
  "_id" : <num>,
  "host" : <hostname:port>,
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 0,
  "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 0
}
```




- **Replica set - Arquitecturas mínimas recomendadas**
 - Primário com dois secundários.

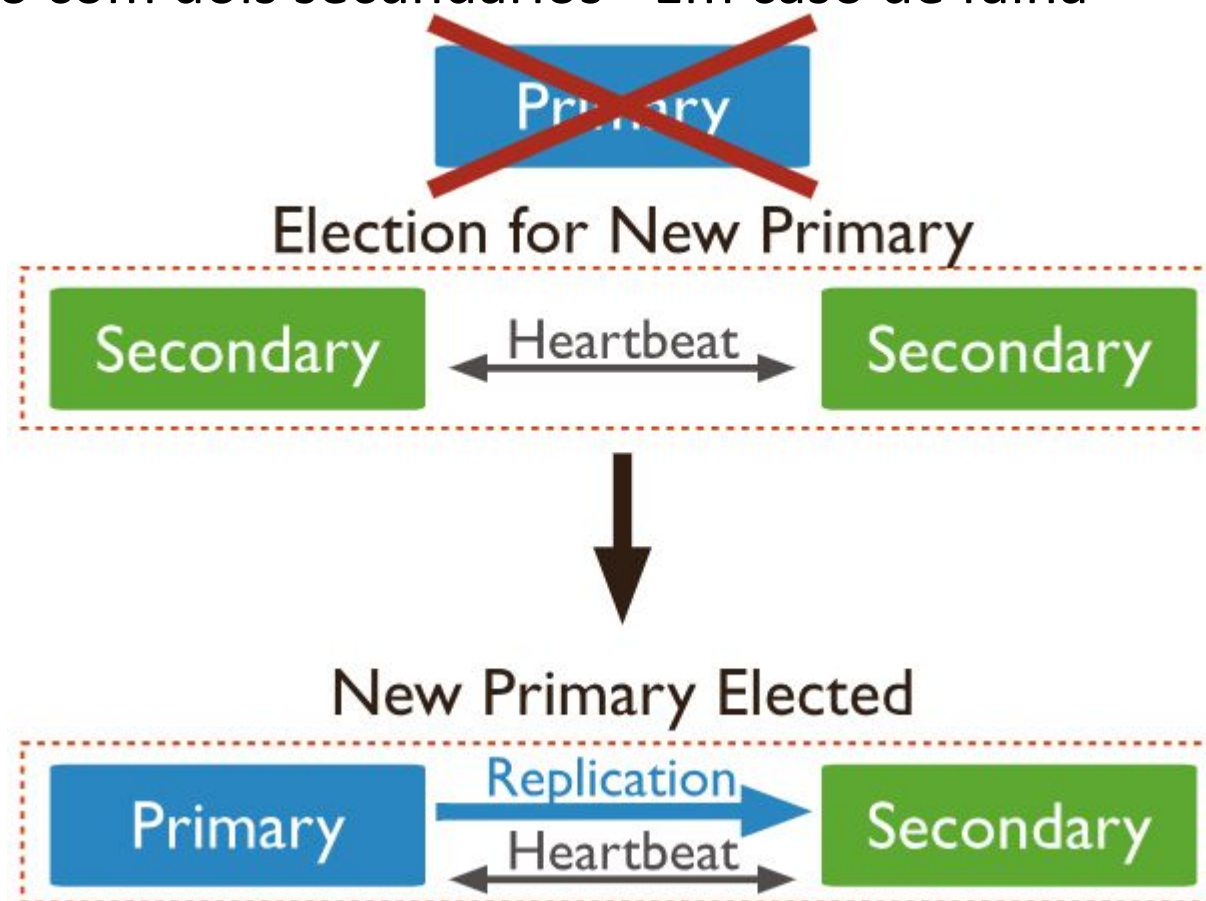




mongoDB®

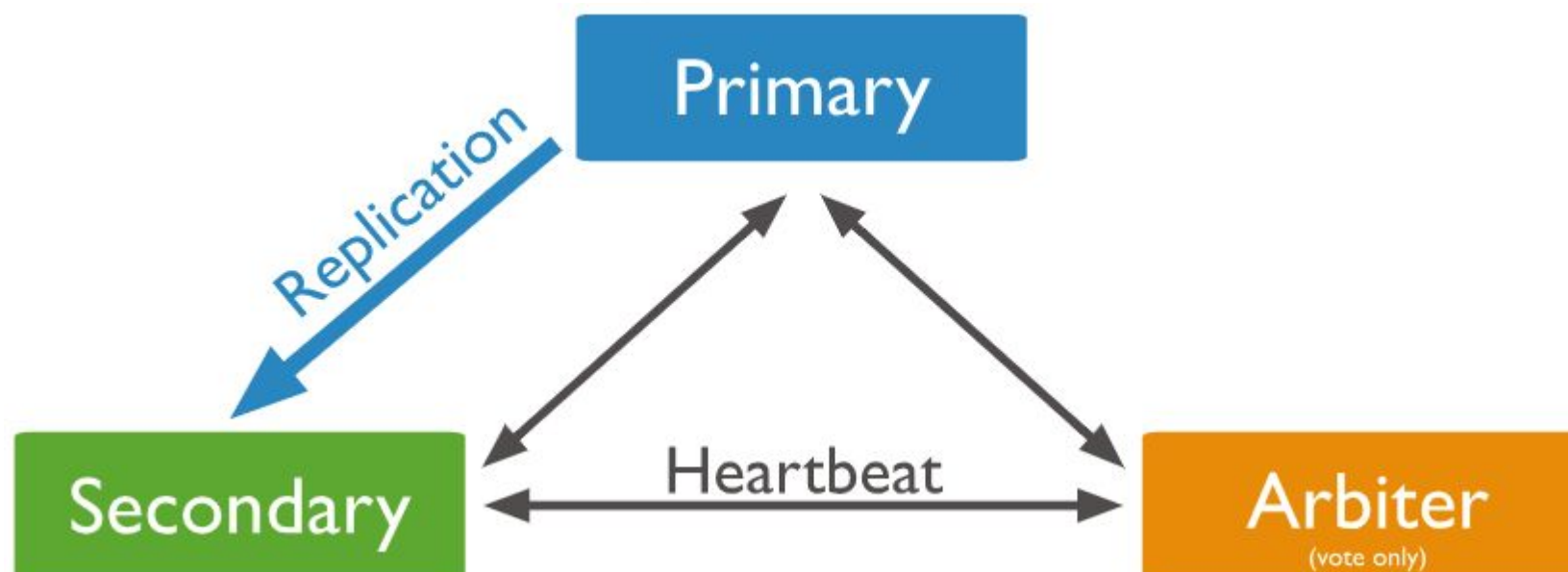
UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Replica set - Arquitecturas mínimas recomendadas**
 - Primário com dois secundários - Em caso de falha





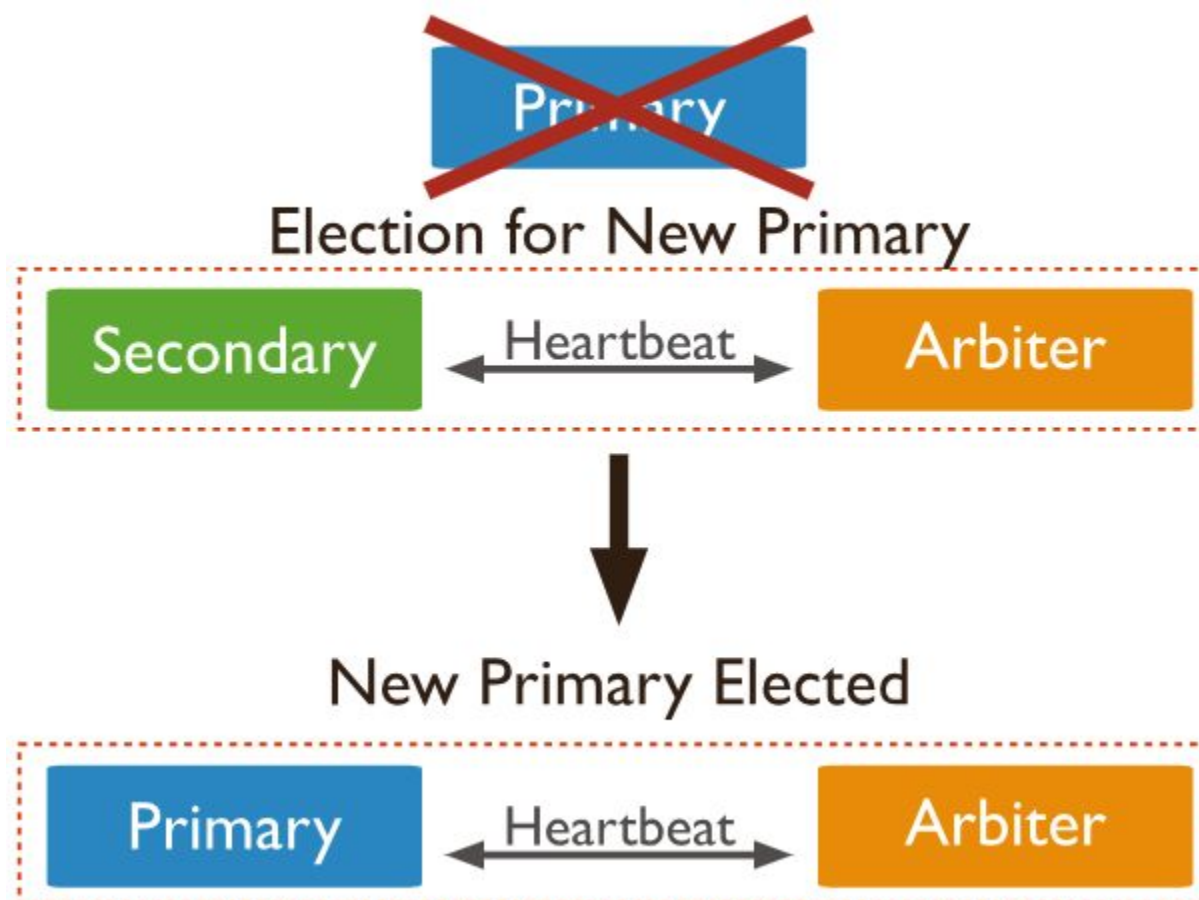
- **Replica set** - Arquitecturas mínimas recomendadas
 - Primário com um secundário e um árbitro.





UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Replica set - Arquitecturas mínimas recomendadas**
 - Primário com um secundário e um árbitro - Em caso de falha

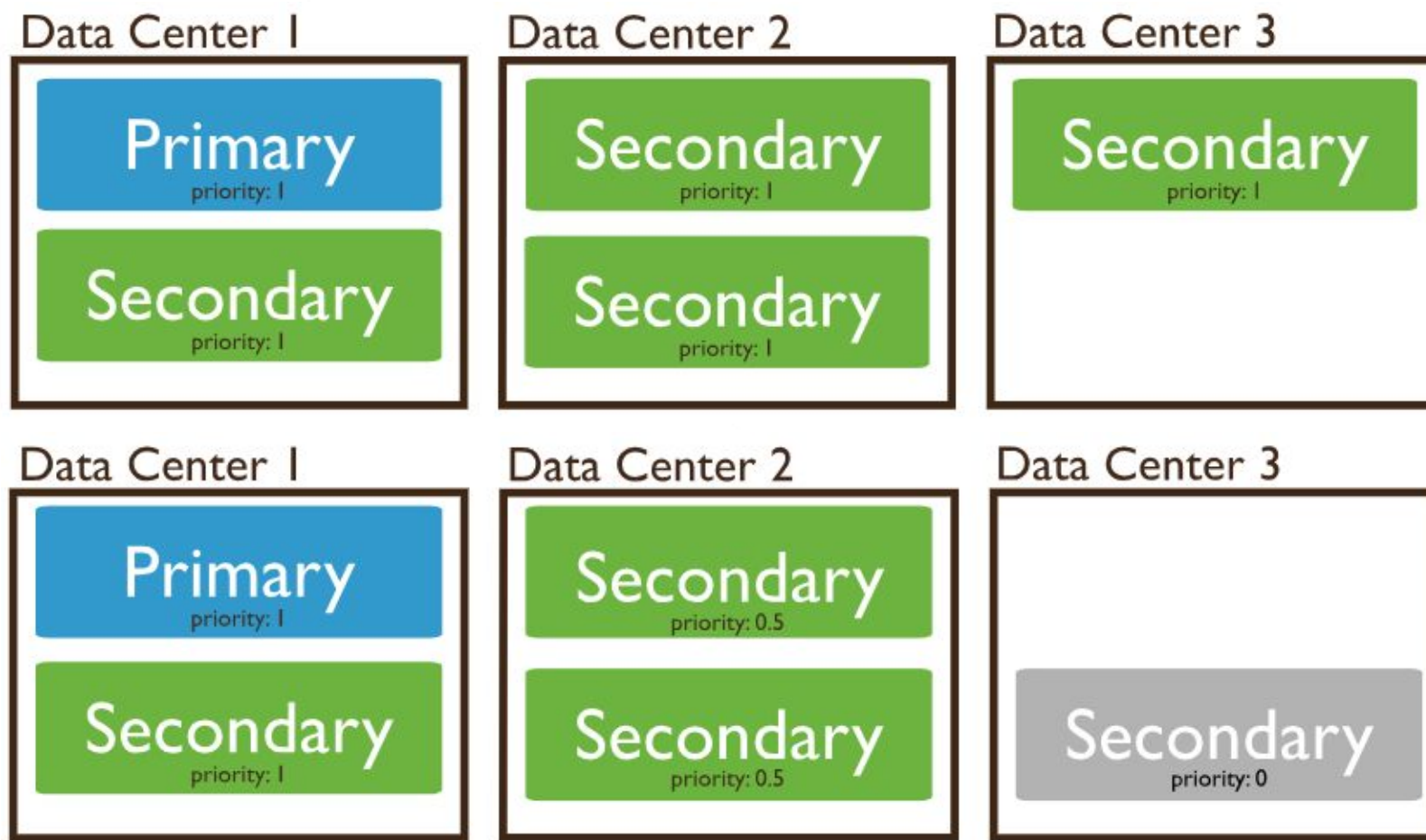




mongoDB®

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Replica set - Arquitecturas *datacenters* recomendadas**





UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Replica set - Exemplo de alguns comandos**

- Criar uma instância do mongo:
 - `mongod --port 27017 --dbpath /var/lib/mongodb --replSet rs0`

- Utilizado dentro do *cliente* mongo:

- Iniciar um *replica set*:
 - `rs.initiate()`
- Adicionar um membro ao *set*:
 - `rs.add(<hostname:port>)`
- Adicionar um árbitro ao *set*:
 - `rs.addArb(<hostname:port>)`
- Ver o estado de um *set*:
 - `rs.status()`
- Remover um membro do *set*:
 - `rs.remove(<hostname:port>)`



mongoDB®

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Replica set - Exemplo de alguns comandos**
 - Utilizado dentro do *cliente* mongo:
 - Verificar se a instância do *mongod* é o primário:
 - `db.isMaster()`
 - Desligar a instância do *mongod*:
 - `db.shutdownServer()`
 - Torna o membro do *replica set* actual inelegível para se tornar primário durante o período especificado:
 - `rs.freeze(seconds)`
 - Instrui o primário do *replica set* para tornar-se secundário. Após os primeiros passos, os secundários elegíveis realizam uma eleição para o primário:
 - `rs.stepDown(stepDownSecs, secondaryCatchUpPeriodSecs)`



mongoDB®

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Replica set - Exemplo de alguns comandos**
 - Utilizado dentro do *cliente* mongo do primário:
 - Definir prioridades para um dos membros ser o eleito o primário:
 - `cfg = rs.conf()`
`cfg.members[0].priority = 0.5`
`cfg.members[1].priority = 0.5`
`cfg.members[2].priority = 1`
`rs.reconfig(cfg)`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- Referências

- <https://docs.mongodb.com/manual/replication/>
- <https://docs.mongodb.com/manual/core/replica-set-members/>



CTeSP

CURSOS TÉCNICOS
SUPERIORES PROFISSIONAIS