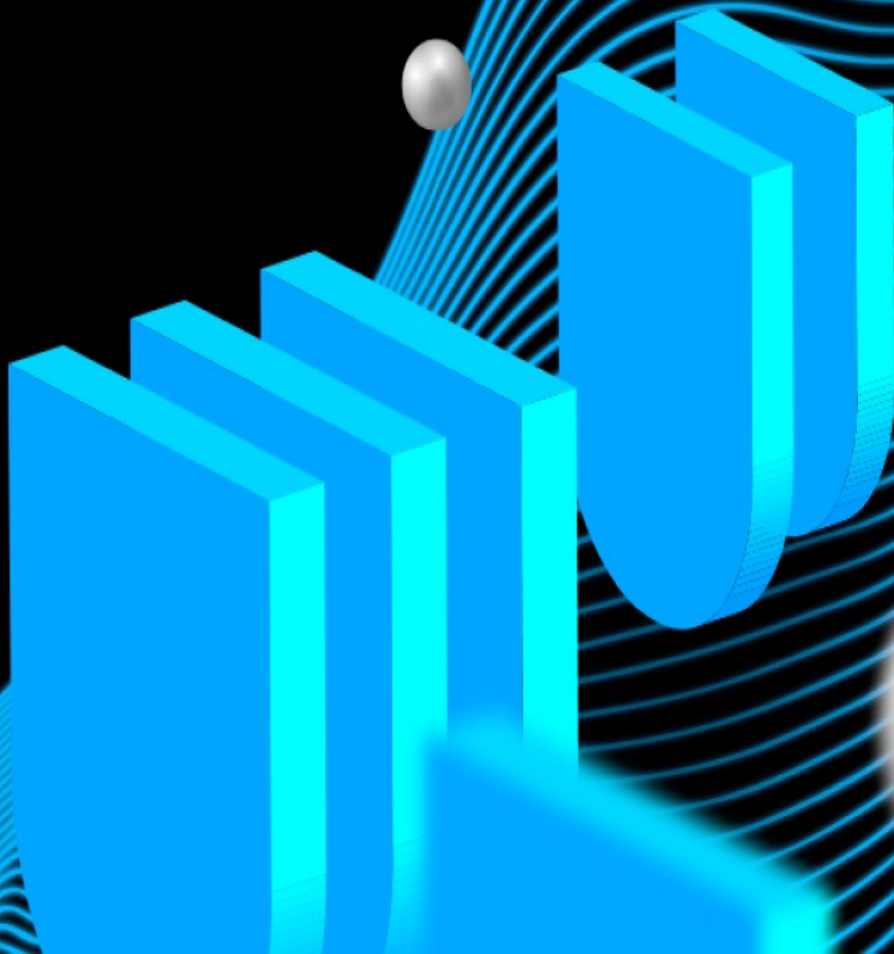


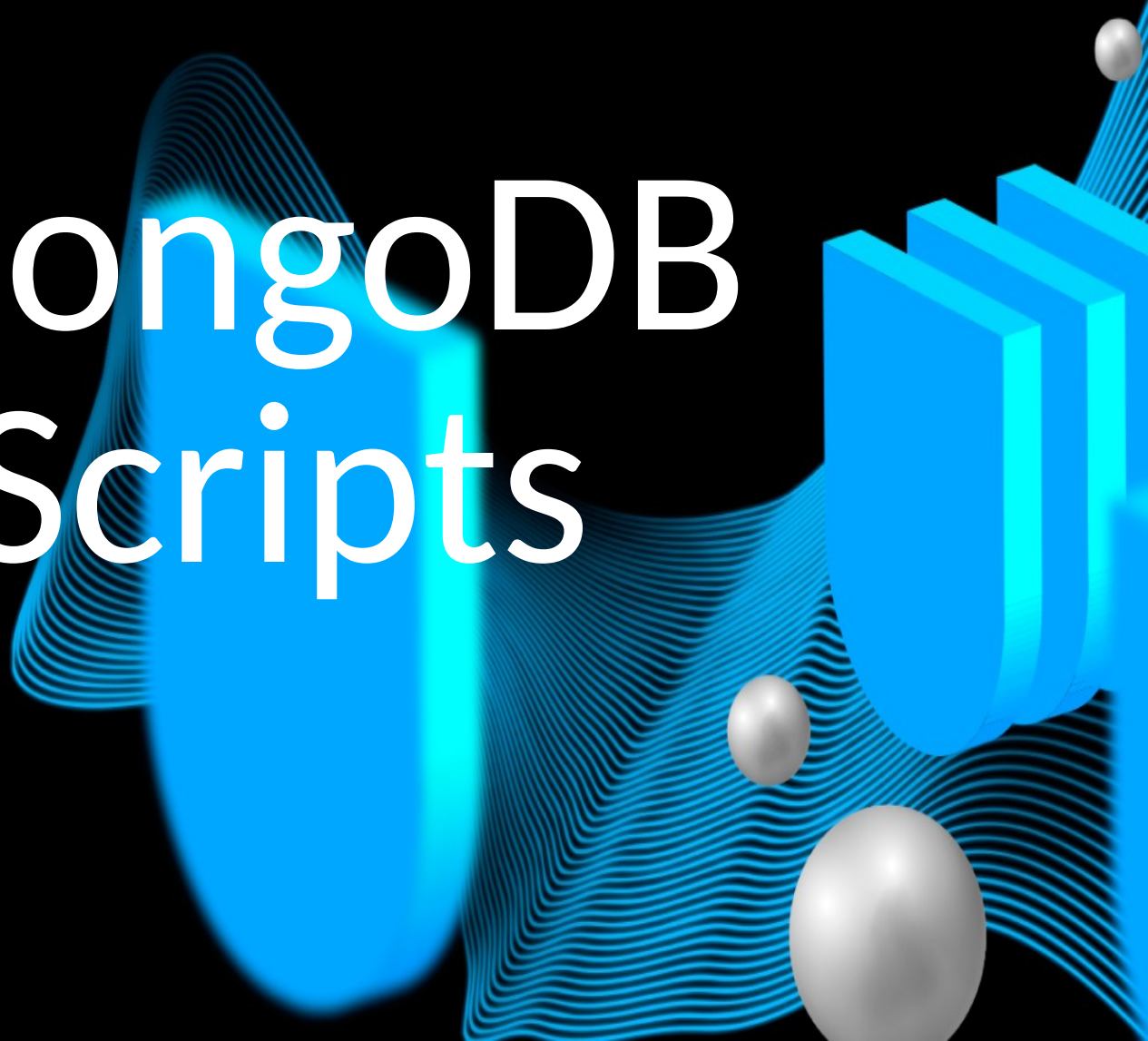


**UNITAU**  
Universidade de Taubaté





# MongoDB Scripts



# Scripts



**O shell do MongoDB permite o uso de scripts em JavaScript para acessar o MongoDB.**

```
> var cols = db.getCollectionNames()
```

```
> print(cols)
```

acervo, alunos, contatos, editora, livro, megasena, produtos, teste



# Scripts



**É possível passar um JavaScript diretamente na linha de comando para ser executado pelo shell.**

# Scripts



```
mongo -u aluno -p aluno --authenticationDatabase  
aluno aluno --eval  
"printjson(db.getCollectionNames())"
```

MongoDB shell version v3.6.2

connecting to: mongodb://127.0.0.1:27017/aluno

MongoDB server version: 3.6.2

```
[  
    "acervo",  
    "alunos",  
    "contatos",  
    "editora",  
    "livro",  
    "megasena",  
    "produtos",  
    "teste"  
]
```

# Scripts



Também é possível executar um arquivo JavaScript diretamente da linha de comando.

**teste.js:**

```
var cols = db.getCollectionNames()  
print(cols)
```

```
mongo -u aluno -p aluno --authenticationDatabase  
aluno aluno teste.js
```

MongoDB shell version v3.6.2

connecting to: mongodb://127.0.0.1:27017/aluno

MongoDB server version: 3.6.2

acervo,alunos,contatos,editora,livro,megasena,produtos,teste



# Scripts



É possível executar um arquivo JavaScript dentro do shell do MongoDB com o comando **load()**.

```
> load("teste.js")
```

```
acervo,alunos,contatos,editora,livro,megasena,produtos,teste  
true
```

# db



**A variável global `db` pode ser usada nos scripts para acessar os métodos dos bancos de dados. Os comandos do shell são uma forma de acessar esses métodos ( shell helpers ) e não podem ser usados nos scripts onde devem ser usados diretamente os métodos.**

**Alguns dos métodos correspondentes aos comandos são:**

Comando	Método
show dbs, show databases	db.adminCommand('listDatabases')
use <db>	db = db.getSiblingDB('<db>')
show collections	db.getCollectionNames()
show users	db.getUsers()



# getCollectionNames



```
> show collections
```

acervo

contatos

editora

livro

megasena

produtos

teste

# getCollectionNames



```
> db.getCollectionNames()  
[  
  "acervo",  
  "contatos",  
  "editora",  
  "livro",  
  "megasena",  
  "produtos",  
  "teste"  
]
```

# adminCommand



```
mongo -u root -p mongoadmin --  
authenticationDatabase admin  
MongoDB shell version v3.6.8  
connecting to: mongodb://127.0.0.1:27017  
MongoDB server version: 3.6.8  
> show databases  
admin      0.000GB  
aluno      0.000GB  
config     0.000GB  
local      0.000GB
```



# adminCommand



```
> db.adminCommand('listDatabases')
{
  "databases" : [
    {
      "name" : "admin",
      "sizeOnDisk" : 176128,
      "empty" : false
    },
    {
      "name" : "aluno",
      "sizeOnDisk" : 528384,
      "empty" : false
    },
  ],
}
```

# adminCommand



```
{  
    "name" : "config",  
    "sizeOnDisk" : 12288,  
    "empty" : false  
},  
{  
    "name" : "local",  
    "sizeOnDisk" : 155648,  
    "empty" : false  
}  
],  
"totalSize" : 872448,  
"ok" : 1  
}
```

# getSiblingDB



A variável **db** permite acesso ao banco de dados corrente. O método **db.getSiblingDB()** permite acesso a outros banco de dados sem alterar o banco de dados corrente.

**db.getSiblingDB(<database>)**

```
> db.getName()
```

```
test
```

```
> dbAluno = db.getSiblingDB('aluno')
```

```
aluno
```



# getSiblingDB



```
> dbAluno.getUsers()  
[  
  {  
    "_id" : "aluno.aluno",  
    "user" : "aluno",  
    "db" : "aluno",  
    "roles" : [  
      {  
        "role" :  
        "db" : "aluno"  
      }  
    ],  
    "readWrite",  
  }  
]
```

# getSiblingDB



```
> db.getName()
```

```
test
```

```
> dbAdmin = db.getSiblingDB('admin')
```

```
admin
```

# getSiblingDB



```
> dbAdmin.getUsers()  
[  
  {  
    "_id" : "admin.root",  
    "user" : "root",  
    "db" : "admin",  
    "roles" : [  
      {  
        "role" :  
"userAdminAnyDatabase",  
        "db" : "admin"  
      }  
    ]  
  }  
]
```