

# ArangoDB



1

## Três Versões

Enterprise, Oasis e Community.

2

## Principais Vantagens

Consolidação, performance simplificada, redução de complexidade, tolerância a falhas.

3

## Principais Serviços

Replicação Datacenter-Datacenter, integração LDAP, backups automatizados, entre outros.

# MongoDB



1

## Três Versões

Clusters compartilhados, clusters dedicados, clusters dedicados multirregionais.

2

## Principais Vantagens

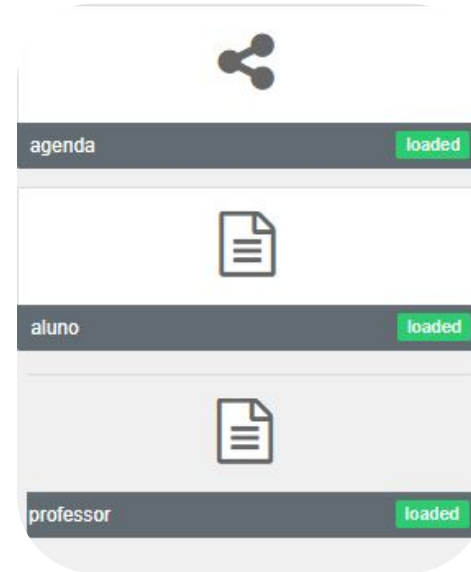
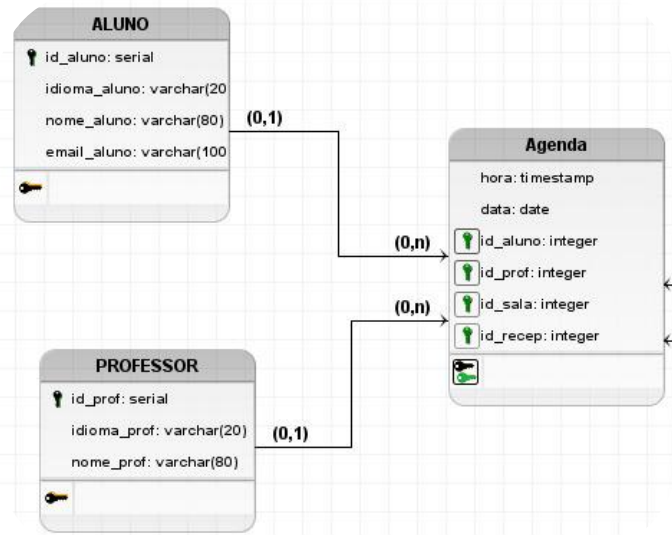
Seguro para dados confidenciais, projetado para a produtividade desenvolvedor.

3

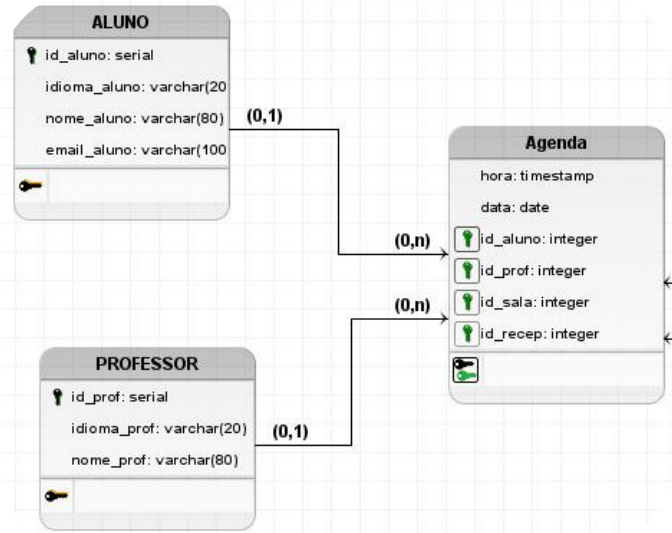
## Principais Serviços

Monitoramento e alertas, cópia de segurança, triggers sem servidor, entre outros.

# Modelo Lógico vs ArangoDB



# Modelo Lógico vs MongoDB



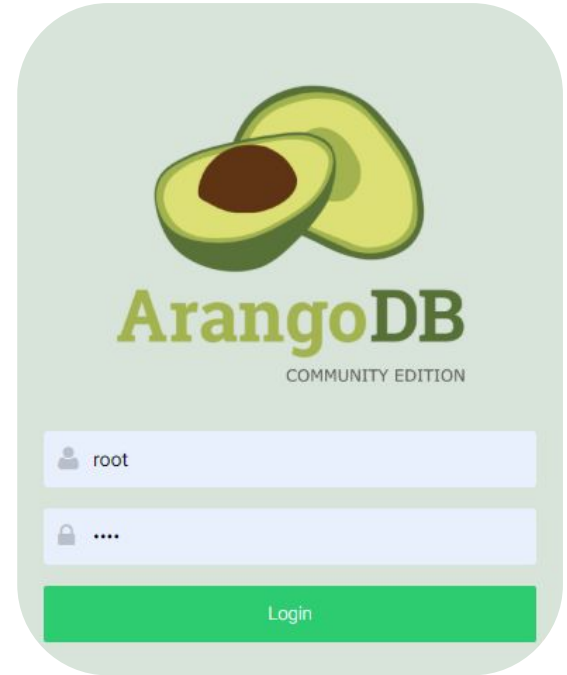
```
{  
  hora:"2020-08-06 15:00:00",  
  data:"2020-08-06",  
}
```

```
ALUNO:{  
  nome_aluno:"Julia",  
  idioma_aluno: "inglês"  
  email: 0"julia@gmail.com"  
},  
PROFESSOR:{  
  nome_prof:"Jorge",  
  idioma_prof:"inglês"  
}  
}
```

# Conexão ArangoDB

---

```
arangosh> db._createDatabase("_system");  
arangosh> var users = require("@arangodb/users");  
arangosh> users.save("root@_system", "1234");  
arangosh> users.grantDatabase("root@_system", "_system");  
shell> arangosh --server.username "root@_system" --server.database _system
```



# Conexão MongoDB

---



Log in to your account



or

Email Address ⓘ

alencarluiza15



alencarluiza15@hotmail.com

Next

Don't have an account? [Sign Up](#)

# Inserção ArangoDB

## Sintaxe

```
For x in [  
{"_nome": "Luis Felipe", "_email": "luis@gmail.com"}] insert x in teste1
```

## Exemplo

```
For alu in [  
{"_nome_aluno": "Carol", "_email_aluno": "carol@gmail.com", "_idioma_aluno": "inglês"},  
] insert alu in aluno
```

```
For prof in [  
{"_nome_prof": "Jorge", "_idioma_prof": "inglês"}] insert prof in Professor
```

```
For ag in [  
{"_hora": "15:00:00", "_data": "2020-07-30", "_from": "aluno/10871", "_to": "professor/6847"}] insert ag in agenda
```

```
...  
{"_data": "2020-07-30", "_from": "aluno/10871", "_hora": "15:00:00", "_id": "agenda/21015", "_key": "21015", "_rev": "_bMDckHa---",  
...  
{"_data": "2020-07-30", "_from": "aluno/10872", "_hora": "15:00:00", "_id": "agenda/21016", "_key": "21016", "_rev": "_bMDckHa--A",  
...  
{"_data": "2020-07-30", "_from": "aluno/10873", "_hora": "15:00:00", "_id": "agenda/21017", "_key": "21017", "_rev": "_bMDckHa--C",  
...  
{"_data": "2020-07-30", "_from": "aluno/10874", "_hora": "15:00:00", "_id": "agenda/21018", "_key": "21018", "_rev": "_bMDckHa--E",  
...  
{"_data": "2020-07-30", "_from": "aluno/10875", "_hora": "16:00:00", "_id": "agenda/21019", "_key": "21019", "_rev": "_bMDckHa--G",  
...  
{"_data": "2020-07-30", "_from": "aluno/10876", "_hora": "16:00:00", "_id": "agenda/21020", "_key": "21020", "_rev": "_bMDckHa--I",  
...  
{"_data": "2020-07-30", "_from": "aluno/10877", "_hora": "17:00:00", "_id": "agenda/21021", "_key": "21021", "_rev": "_bMDckHa--K",  
...  
{"_data": "2020-07-30", "_from": "aluno/10878", "_hora": "17:00:00", "_id": "agenda/21022", "_key": "21022", "_rev": "_bMDckHa--M",  
...  
{"_data": "2020-08-03", "_from": "aluno/10879", "_hora": "15:00:00", "_id": "agenda/21023", "_key": "21023", "_rev": "_bMDckHa--O",  
...  
{"_data": "2020-08-03", "_from": "aluno/12034", "_hora": "15:00:00", "_id": "agenda/21024", "_key": "21024", "_rev": "_bMDckHi---",  
...
```

# Inserção MongoDB

```
db.aluno.insert(
  {nome_aluno: "Julia", idioma_aluno: "inglês", email:
    "julia@gmail.com"},
  {nome_aluno: "Gabriela", idioma_aluno: "francês", email:
    "gabriela@gmail.com"},
  {nome_aluno: "Rafael", idioma_aluno: "espanhol", email:
    "rafael@gmail.com"},
  {nome_aluno: "Elisa", idioma_aluno: "inglês", email:
    "elisa@gmail.com"}
)
```

## Insert to Collection

VIEW {} ≡

```
1 {
2   "_id": {
3     "$oid": "5f7610788e7f82a0c033dddc"
4   },
5   "nome_aluno": "Elisa",
6   "idioma_aluno": "inglês",
7   "email": "elisa@gmail.com"
8 }
```

## Escola.Aluno

COLLECTION SIZE: 301B TOTAL DOCUMENTS: 3 INDEXES

Find

Indexes

Schema Anti-Patterns ⓘ

FILTER {"filter": "example"}

QUERY RESULTS 1-4 OF 4

```
_id: ObjectId("5f7608418e7f82a0c033ddcf")
nome_aluno: "Gabriela"
idioma_aluno: "francês"
email: "gabriela@gmail.com"
```

```
_id: ObjectId("5f7608818e7f82a0c033ddd0")
nome_aluno: "Julia"
idioma_aluno: "inglês"
email: "julia@gmail.com"
```

```
_id: ObjectId("5f7608a38e7f82a0c033ddd2")
nome_aluno: "Rafael"
idioma_aluno: "espanhol"
email: "rafael@gmail.com"
```

```
_id: ObjectId("5f7610788e7f82a0c033dddc")
nome_aluno: "Elisa"
idioma_aluno: "inglês"
email: "elisa@gmail.com"
```

## Escola.Agenda

COLLECTION SIZE: 186B TOTAL DOCUMENTS: 3 INDEXES

Find

Indexes

Schema Anti-Patterns

FILTER {"filter": "example"}

QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("5f76115a8e7f82a0c033ddd")
_id_aluno: "Object_Id(5f7608a38e7f82a0c033ddd2)"
_id_prof: "Object_Id(5f76092d8e7f82a0c033ddd4)"
hora: "15:00:00"
data: "2020-08-06"
```

```
_id: ObjectId("5f7612258e7f82a0c033ddd")
_id_aluno: "Object_Id(5f7608418e7f82a0c033ddcf)"
_id_prof: "Object_Id(5f76094e8e7f82a0c033ddd6)"
hora: "16:00:00"
data: "2020-08-06"
```

```
_id: ObjectId("5f7612448e7f82a0c033ddf")
_id_aluno: "Object_Id(5f7608818e7f82a0c033ddd0)"
_id_prof: "Object_Id(5f7609668e7f82a0c033ddd7)"
hora: "17:00:00"
data: "2020-08-06"
```



# Seleção ArangoDB

---

```
for alu in aluno
  for prof in professor
    for ag in agenda
      filter ag.from == alu.id
      filter ag.to == prof.id
```

```
return distinct {
  Aluno: alu,
  Professor: prof,
  Agenda: ag
}
```

# Seleção MongoDB

db.aluno.find()  
db.professor.find()  
db.agenda.find()

## Escola.Professor

COLLECTION SIZE: 277B    TOTAL DOCUMENTS: 4    INDEXES TOTAL SIZE: 36KB

**Find**

Indexes

Schema Anti-Patterns

Aggregation

Search Indexes

INSERT DOCUMENT

**FILTER** {"filter": "example"}

**Find**

**Reset**

# Exclusão

# ArangoDB

---



```
LET doc = DOCUMENT('professor/7045')  
REMOVE doc IN professor
```

```
LET doc = DOCUMENT('professor/6939')  
REMOVE doc IN professor
```

# Exclusão

## MongoDB

---



```
db.aluno.remove({ nome_aluno: {$gt:"Elisa" } })
```

```
db.agenda.remove({ hora: {$gt:"17:00:00" } })
```



```
_id: ObjectId("5f7608f68e7f82a0c033ddd3")  
nome_prof: "Maristela"  
idioma_prof: "inglês"
```



# Atualização ArangoDB

```
LET key = PARSE_IDENTIFIER("professor/6847").key  
UPDATE key WITH { _idioma_prof: "francês" } IN professor
```

```
LET key = PARSE_IDENTIFIER("aluno/10871").key  
UPDATE key WITH { _idioma_aluno: "francês" } IN aluno
```

# Atualização MongoDB

```
db.agenda.update( { hora: { $gt: "15:00:00" } }, { $set: {  
  data: "2020-08-20" } }, { multi: true } )  
db.agenda.update( { hora: { $gt: "17:00:00" } }, { $set: {  
  data: "2020-08-15" } }, { multi: true } )
```



```
_id: ObjectId("5f7608f68e7f82a0c033ddd3")  
nome_prof: "Maristela"  
idioma_prof: "inglês"
```



```
1 _id: ObjectId("5f7608f68e7f82a0c033ddd3")  
2 nome_prof : "Maristela "  
3 idioma_prof : "inglês "
```

ObjectId  
String  
String

CANCEL

UPDATE

# Pontos Positivos e Negativos do ArangoDB

1. Linguagem AQL é bem simples de entender.
2. Documentação oficial é boa e fácil de encontrar.
3. Por ser um pouco conhecido, tem vários vídeos no Youtube de tutoriais.
4. Por ter a versão *community*, fica melhor de testar a aplicação antes de adquirir a versão paga.



1. Quando precisa fazer joins com várias tabelas, não é muito bom, vários valores vêm repetidos e é difícil visualizá-los.



# Pontos Positivos e Negativos do MongoDB

1. Quando os dados não se encaixam no modelo relacional;
2. Quando seus dados são na verdade objetos;
3. Quando sua aplicação valida a consistência dos dados, não o banco;
4. Quando o projeto da aplicação é orientado ao domínio;
5. Quando seu projeto precisa se adaptar a mudanças;
6. Quando a escalabilidade é importante;
7. Quando inovação não é um problema.



1. Ao manusear o MongoDB Atlas, não foi o esperado para conseguir programar, pois o site é bem difícil de achar as informações que precisa. Ao procurar os tutoriais, existem vários jeitos de programar, mas nenhum foi possível executar no MongoDB Atlas, e nem no Colab. O único caminho foi seguindo as instruções de encaixar as informações que precisava no site, ele foi me conduzindo, mas não era o meu esperado para executar os códigos.







## Comparação entre ArangoDB e MongoDB

---

### MongoDB

Como o MongoDB tem duas versões, baixado e o site, é possível programar pelo prompt de comando quando o programa é instalado. Mas na interface dele, é impossível.

Não foi possível encontrar aonde se programa, só tem apenas a direção que o próprio programa te dá, de como adicionar o Database e as Collections, e assim inserindo os dados, como também te permite alterar ou excluir. Pelo comando, é possível usar os códigos, mas com a falta de algumas informações não foi possível continuar. E no site MongoDB Atlas, percebi que é a mesma coisa que o programa baixado, ele te direciona o que fazer e o que colocar para criar e inserir. Usando pelo Colab, não foi possível usar os códigos, pois a maioria não funcionava, e ao procurar informações, percebi que existem diversos tipos de códigos, cada um de forma diferente, mas com a mesma função, exemplo:

```
.insert()  
.insert_one()  
.insert_many()
```

Assim como todos os outros.



## Comparação entre ArangoDB e MongoDB

---

### ArangoDB

Na versão community do ArangoDB inclui a interface web, facilitando assim a criação e os testes nas coleções.

A documentação oficial é boa, fácil de achar e bem explicada, tanto para a instalação quanto para a linguagem utilizada para manipular os dados.

Como um banco de dados multi-model, o ArangoDB minimiza componentes para se manter, reduzindo a complexidade das tecnologias que uma aplicação utiliza.

Flexível e com suporte a linguagens de programação (como Python) e por ser uma solução completa e livre.

O ArangoDB é um SGBD que trabalha com grafos porque permite armazenar vértices e arestas com propriedades, e oferece funcionalidades para o uso de grafos.



## **Comparação entre ArangoDB e MongoDB**

---

Chegamos a conclusão que o ArangoDB ser melhor, devido a sua facilidade de programação, o fácil acesso a documentação e por ter uma interface web que auxilia na criação e nas consultas das coleções.

# Principais Referências

---

## **ArangoDB**

<https://www.arangodb.com/docs/stable/>

<https://www.arangodb.com/docs/stable/aql/>

<https://youtu.be/4C4zqhXwCKs>

## **MongoDB**

<https://www.mongodb.com/cloud/atlas>

<https://medium.com/leroy-merlin-brasil-tech/devo-usar-nosql-e-mongodb-951693aa0d34>

<https://docs.mongodb.com/manual/reference/method/db.collection.remove/>